

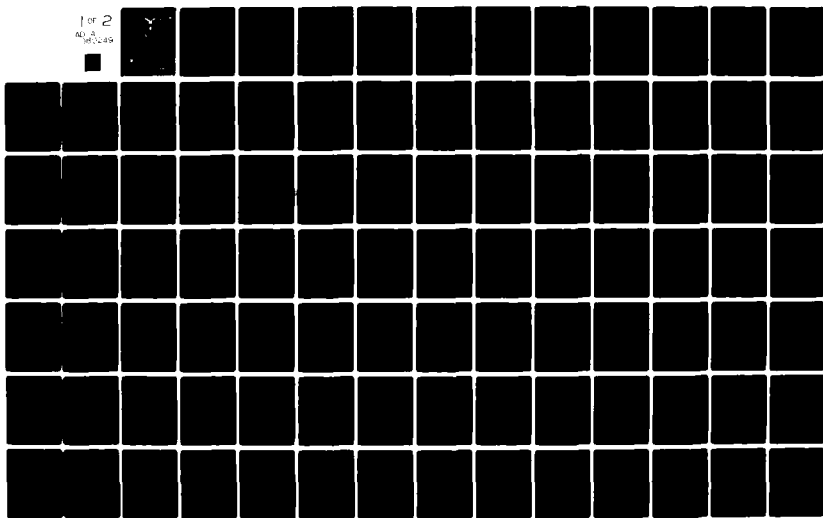
AD-A086 249

AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCH00--ETC F/6 17/5
AN ADAPTIVE DISTRIBUTED-MEASUREMENT EXTENDED KALMAN FILTER FOR --ETC(U)
DEC 79 R L JENSEN, D A HARNLY
AFIT/6A/EE/79-1-VOL-1

UNCLASSIFIED

NL

1 of 2
AD-A
1972-49



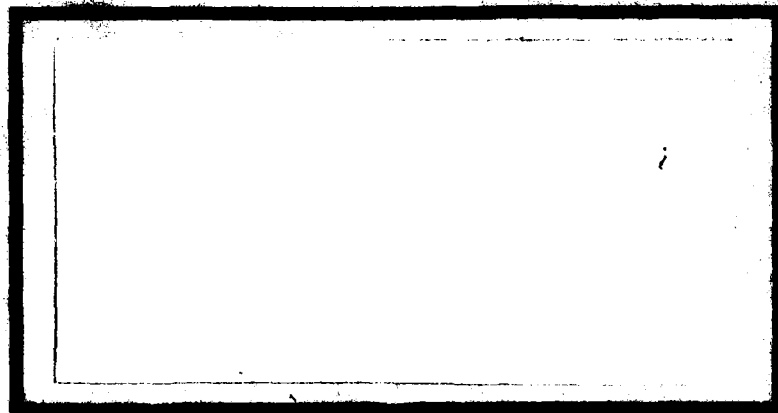
ADA080249



LEVEL



DDC FILE COPY



UNITED STATES AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DDC
RECEIVED
FEB 5 1960
RECEIVED

A

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

AFIT/GA/EE/79-1

A086238

A

AN ADAPTIVE DISTRIBUTED-MEASUREMENT
EXTENDED KALMAN FILTER FOR
A SHORT RANGE TRACKER (I)

THESIS

AFIT/GA/EE/79-1

Vol I

Robert L. Jensen
Capt USAF

Douglas A. Harnly
Capt USAF

Approved for public release; distribution unlimited

(14) AFIT/GA/EE/79-1-VOL-1

(6) AN ADAPTIVE DISTRIBUTED-MEASUREMENT EXTENDED
KALMAN FILTER FOR A SHORT RANGE TRACKER,
VOLUME I.

(9) Master's THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

(10) Robert L. Jensen

Capt

USAF

and
Douglas A. Harnly

Capt

USAF

Graduate Astronautical Engineering

(11) December 1979

(12) 173

Approved for public release; distribution unlimited.

012225

LB

Preface

This study was part of a continuing effort to design a missile tracker for one of the Air Force Weapons Laboratory's laser weapons using modern estimation techniques. An adaptive extended Kalman Filter which uses direct infrared sensor data was synthesized and tested. We hope that this effort brings AFWL closer to an online realization of such a filter.

We wish to express our thanks to Captain Z. H. Lewantowicz of AFWL for his advice and cooperation in this project. A special thanks goes to our advisor Dr. Peter S. Maybeck for his motivation, advice and time. Finally, an inadequate thank you to our wives, Pat and Debbie, for their understanding and support.

Robert L. Jensen

Douglas A. Harnly

Volume I

Contents

	Page
Preface	ii
List of Figures	vi
List of Tables	ix
List of Symbols	xi
Abstract	xv
I. Introduction	1
Background	1
Problem	2
Research Outline	3
Assumptions	4
FLIR	4
Target	5
Intensity Pattern	6
Background Noise	8
Closed Loop	8
Previous Investigation	8
Truth Model Changes	12
II. Truth Model Development	14
Introduction	14
Spatially Correlated Background Noise	14
Elliptical Intensity Contour Target	23
Target Dynamics Model	28
Temporally Correlated Background Noise	37
Field Size	41
REFERENCING	42
III. Previous Filter Robustness	45
Introduction	45
Sensitivity to Parameter Mismatches	46
Sensitivity to New Truth Model	55
Major Problem Areas	66
IV. Filter Requirements	69
Introduction	69
Six State Filter	70
Eight State Filter	74
Size and Shape Estimation	77

Volume I

Contents

	Page
Dynamic Driving Noise Estimation	83
I_{\max} Estimation	87
V. Adaptive Filter Performance and Modifications .	89
Introduction	89
I_{\max} Estimation	89
AR and σ Estimation	98
Target Acquisition Procedures	103
Performance of Target Dynamics Equations .	108
QFD Estimation Performance Analysis	122
Maneuver Required Modifications	125
Final Filter Performance	136
VI. Conclusions and Recommendations	146
Conclusions	146
Recommendations	148
Computer Support	151
Bibliography	152
Vita	154

Volume II

Contents

	Page
List of Figures	iii
List of Tables	v
List of Symbols	vi
Appendix A: Previous Truth Model and Filter Equations	1
Appendix B: Real Data Analysis for Spatial Corre- lations of Background Noise	11
Appendix C: Real Data Analysis for Temporal Corre- lations of Background Noise	17
Appendix D: Q_{Fd} Calculation	21
Appendix E: Computer Simulation Considerations	26
Appendix F: FORTRAN Code: Main Programs and Library of Subroutines	31
Appendix G: FORTRAN Code: Plot Programs	71
Appendix H: Input Parameters for Cases 1-31	79
Appendix I: Performance Plots for Cases 1-12	85
Appendix J: Performance Plots for Cases 13-31	144
Appendix K: Input Parameters for Cases 32-37	215
Appendix L: Performance Plots for Cases 32-37	217

List of Figures

Figure		Page
1	Gaussian Bivariate Distribution Intensity Model, $I(x,y)$	6
2	Target Image in FLIR Field of View	7
3	Pixel Numbering Scheme	19
4	First Neighbor Correlation Coefficients	19
5	First and Second Neighbor Correlations	20
6	Image Intensity Characteristics	24
7	Image Projection	25
8	Azimuth Geometry	32
9	Elevation Geometry	33
10	Trajectory 1: Approaching Target	34
11	Trajectory 2: Cross Range Target	35
12	Trajectory 3: 20-g Turn	36
13	Continuous First Order Noise Generation Model	38
14	Discrete First Order Noise Generation Model	39
15	Discrete First Order Noise Generation Model Driven by Unit Variance Noise	39
16	Case 2- Target Position: Standard Deviation (σ ,Sigma) Convergence	48
17	Case 2- Target Position: Real Error versus Filter Standard Deviation	49
18	Case 2- Target Position: Mean Error and 1 Sigma Envelope	50
19	Image Geometry Characteristics	77
20	I_{\max} Estimation: Maximum Measurement	91

Figure		Page
21	I_{\max} Estimation: Averaged Maximum Measurement	92
22	Average Value of Gaussian Curve Over Interval 0 to 1 (1 Pixel)	93
23	I_{\max} Estimation: Maximum Measurement with Error Function	96
24	I_{\max} Estimation: Averaged Maximum Measurement with Error Function	97
25	Transient Response of σ_v and AR Estimation	99
26	Steady State Response of σ_v and AR Estimation	100
27	Truth Model and Filter Image Comparison . .	101
28	σ_v and σ_{pv} for AR = 3 (Truth Model)	102
29	σ_v and σ_{pv} for AR = 1 (Truth Model)	103
30	Case 32: Target Position Mean Error and 1 Sigma Envelope - x channel	106
31	Case 32: Target Velocity Mean Error and 1 Sigma Envelope - x channel	107
32	Case 33: Target Position Mean Error and 1 Sigma Envelope - x channel	110
33	Case 33: Target Position Mean Error and 1 Sigma Envelope - y channel	111
34	Case 33: Target Velocity Mean Error and 1 Sigma Envelope - x channel	112
35	Case 33: Target Velocity Mean Error and 1 Sigma Envelope - y channel	113
36	Case 34: Target Position Mean Error and 1 Sigma Envelope - x channel	115
37	Case 34: Target Position Mean Error and 1 Sigma Envelope - y channel	116

Figure		Page
38	Case 34: Target Velocity Mean Error and 1 Sigma Envelope - x channel	117
39	Case 34: Target Velocity Mean Error and 1 Sigma Envelope - y channel	118
40	Case 36: Target Position Mean Error and 1 Sigma Envelope - x channel	120
41	Case 36: Target Velocity Mean Error and 1 Sigma Envelope - x channel	121
42	Image/Filter Image Separation by Acceleration	126
43	$\underline{H}^T(\underline{-h})$ versus δ	131
44	Intercept for Equation (156)	132
45	Case 36: Target Position Mean Error and 1 Sigma Envelope - x channel	137
46	Case 36: Target Velocity Mean Error and 1 Sigma Envelope - x channel	139
47	Case 36: Target Position Standard Deviation, Actual and Filter - indicated x channel	140
48	Case 36: Target Velocity Standard Deviation, Actual and Filter - indicated x channel	141
49	Case 36: Target Position Mean Error and 1 Sigma Envelope - y channel	142
50	Case 36: Target Velocity Mean Error and 1 Sigma Envelope - y channel	143
51	Case 36: Target Position Standard Deviation, Actual and Filter - indicated y channel	144
52	Case 36: Target Velocity Standard Deviation, Actual and Filter - indicated y channel	145

List of Tables

Table		Page
I	Baseline for Cases 3-18: Case 2 Results. . .	47
II	Target Dynamics Mismatch - Case 3: Low τ_{DT} and Case 4: High τ_{DT} Results	52
III	Target Intensity Mismatch - Case 5: Low I_{max} and Case 6: High I_{max} Results	53
IV	Target Size Mismatch - Case 7: $\sigma_{DT} = 1$ and Case 8: $\sigma_{DT} = 5$ Results	54
V	Background Noise Mismatch - Case 9: $S/N = 1$ and Case 10: $S/N = 20$ Results	55
VI	Ratio of Atmospheric Jitter to Target Dynamics Mismatch - Case 11: $\sigma_D/\sigma_A = .2$ and Case 12: $\sigma_D/\sigma_A = 5$ Results	56
VII	Elliptical Target Intensity Contours in the Truth Model - Case 13, 14, and 15 Results	58
VIII	Spatially Correlated Background Noise in the Truth Model - Case 17, 18, and 19 Results	59
IX	Baseline for Cases 21-24: Case 20 Results .	61
X	Target Motion in the Truth Model - Case 21 and Case 22 Results	62
XI	Temporal and Spatial Background Noise in the Truth Model - Case 24 Results	63
XII	Baseline for Cases 27-30: Case 26 Results	64
XIII	Temporal and Spatial Noise in the Truth Model ($S/N = 2$): Cases 27, 28, and 30 Results	65
XIV	Temporal and Spatial Background Noise with Target Motion in the Truth Model ($S/N = 2$): Case 31 and Case 32 Results . . .	67

Table		Page
XV	I_F Variation with σ_{vF} for Single Simulations of Cross Range Target ($\sigma_{pv} = 1$ pixel)	94
XVI	Six State Filter Tracking Cross Range Target with and without \underline{Q}_{Fd} Estimation - Case 33 and Case 37	123
XVII	Maneuver Indicators	128
XVIII	Elevation Acceleration Estimate Only . . .	134
XIX	Elevation Acceleration Estimate and Increased \underline{P} Matrix	135

List of Symbols

<u>A</u>	General parameter vector
<u>A_p</u>	Area of picture element
AR	Aspect ratio: $AR = \sigma_v / \sigma_{pv}$
<u>b</u>	Realization of parameter vector
<u>B</u>	System control input matrix
<u>E[.]</u>	Expected value
<u>F</u>	System plant matrix
<u>G</u>	System noise input matrix
<u>H</u>	Linearization of intensity measurements
<u>h()</u>	Nonlinear measurement relation
Hz	Hertz
<u>I_F</u>	Filter maximum intensity
<u>I_{max}</u>	Maximum target intensity
<u>K</u>	Kalman filter gain matrix
<u>L</u>	Likelihood function
<u>L'</u>	Full scale maximum likelihood equation
<u>P</u>	Covariance matrix
<u>P_L</u>	Covariance of likelihood function
<u>Q</u>	Strength of disturbance process matrix
<u>q</u>	Noise strength
<u>R</u>	Covariance of measurement noise matrix
<u>r</u>	Correlation coefficient
<u>r</u>	Residual vector
<u>R_F</u>	Measurement noise variance
S/N	Signal to noise ratio: $S/N = I_{max} / \sigma_N$

List of Symbols

t	Time
\underline{u}	Deterministic velocity input function
\underline{v}	General velocity vector
w	White noise
$\hat{\underline{x}}$	Filter state estimate vector
\underline{x}	State vector
\hat{x}_A	Estimate of horizontal position due to atmospheric jitter
\hat{x}_D	Estimate of horizontal velocity due to target motion
$\hat{\dot{x}}_D$	Estimate of horizontal velocity due to target motion
x_{peak}, x_p	Horizontal coordinate of Gaussian intensity function maximum
\hat{y}_A	Estimate of vertical position due to atmospheric jitter
\hat{y}_D	Estimate of vertical position due to target motion
$\hat{\dot{y}}_D$	Estimate of vertical velocity due to target motion
y_{peak}, y_p	Vertical coordinate of Gaussian intensity function maximum
\underline{z}	General measurement vector
α	Azimuth
β	Elevation
\underline{r}	Vector of actual measurements
γ	Angle between velocity vector and image plane
$\Delta \underline{x}$	Filter state vector update
Δx_v	Image plane velocity (v_{LOS}) direction coordinate of intensity pattern

List of Symbols

Δy_v	Coordinate perpendicular to image plane velocity (v_{LOS}) of intensity pattern
δ	Intensity peak separation due to acceleration
θ	Orientation angle in image plane
ρ	Line of sight range
σ_A	RMS value of atmospheric jitter (pixels)
σ_F	RMS value of FLIR noise
σ_g	Dispersion of Gaussian intensity function
σ_D	RMS value of target motion (pixels)
σ_N	RMS value of background noise
σ_R	Variance of measurement noise
τ_A	Correlation time of atmospheric jitter
τ_D	Correlation time of target dynamics
τ_N	Correlation time of background noise
Φ	State transition matrix

Subscripts

A	Atmospheric jitter
D	Target dynamics
d	Discrete form
F	Filter
h	Horizontal
I	Inertial
o	Initial value
peak	Maximum intensity position
pv	Direction perpendicular to image plane velocity

List of Symbols

T	Truth model
v	Image plane velocity frame
α, az	Azimuth direction
β, el	Elevation direction

Superscripts

\cdot	Time derivative
\wedge	Estimate
$+$	After update
$-$	Before update

Abstract

An adaptive Extended Kalman Filter algorithm is designed to track a distributed (elliptical) source target in a closed loop tracking problem, using outputs from a forward looking infrared (FLIR) sensor as measurements. The filter adaptively estimates image intensity, target size and shape, dynamic driving noise, and translational position changes due to two effects: actual target motion, and atmospheric jitter. Atmospheric backgrounds are studied for the effect of temporal and spatial correlations on filter performance. A Monte Carlo analysis is conducted to determine filter performance for two target scenarios: approximately straight approach and cross range constant velocity. Good performance is obtained for the first two trajectories. For the second trajectory, a one sigma tracking error of .2 pixel (4 μ rad) with a signal to noise ratio of 12.5. The filter adapts well to changes in image intensity, size, and shape.

AN ADAPTIVE DISTRIBUTED-MEASUREMENT
EXTENDED KALMAN FILTER FOR USE IN A
SHORT RANGE TRACKER

I. Introduction

"...I believe that if the difficult technical hurdles facing directed energy technology are surmounted, the application of this technology to military needs may revolutionize both strategic and tactical warfare." (Ref 1:48)

Dr. Ruth M. Davis, Deputy Under
Secretary of Defense for
Research and Advanced Technology

Background

Since its conception and early development, the laser has been desired as a weapon because of its unique characteristics. Chief among these is the speed of light at which energy is transmitted from source to destination. Thus, a laser virtually eliminates time of flight considerations; it is no longer necessary to "lead" the target as other weapon systems, such as radar guided anti-aircraft artillery, must. With a laser, the destructive energy reaches the target essentially instantaneously. Recent technological advances with high energy lasers have made the deposition of large amounts of electromagnetic energy on a small area in short periods of time feasible. This increase in laser technology, combined with speed of light transmission, has attracted the attention of high level military and Department of Defense leaders, such as William J. Perry, Under Secretary of Defense for Research and Engineering (Ref 2:62), and General Alton D. Slay, Commander of the Air Force Systems Command (Ref 3:50), as well as Dr. Ruth B. Davis, Deputy Under Secretary of Defense for Research and Advanced Technology, whose quote begins this paper.

Implementation of this desirable weapon system requires high power for laser pumping, very accurate optics control (pointing), and very accurate target position estimation. The last two requirements are dictated by the need to maintain the laser beam on a specific part of the target long enough to destroy the vehicle or one of its critical components. "Painting" the entire target with laser energy is inefficient and would require extremely high amounts of energy to achieve destruction, and a very accurate pointing and tracking system makes it unnecessary. The focus of this research is to investigate a precise target estimation algorithm for possible use in laser pointing and tracking.

An adaptive extended Kalman filter as a missile tracker has considerable potential to use more information from the sensor measurements than a correlation algorithm can. First, the statistical characteristics of the atmospheric disturbance of the infrared (IR) wavefront are well known and can be modelled in the extended Kalman filter. Thus, separation of the atmospheric jitter from true target motion is possible. This separation is important since the high energy laser wavefront will not undergo the same atmospheric distortion as the infrared wavefronts (Ref 4:1). Second, the extended Kalman filter has the ability to predict future target motion through an internal target dynamics model. An adaptive extended Kalman filter can also provide fast detection of target maneuvering and can estimate uncertain parameters such as descriptors of target size and shape. Finally, as needed, the filter can model the temporal and spatial correlations of background noise, which would enhance the separation of atmospheric jitter and target dynamics.

Problem

The Air Force Weapons Laboratory (AFWL) located at Kirtland Air Force Base, New Mexico, is currently using a Forward Looking Infrared (FLIR) Sensor in conjunction with

a "coarse" correlation tracking algorithm and a fine track Far Field Intensity Measurement (FFIM) system to provide accurate target position estimates. Specifically, AFWL is interested in tracking air-to-air missiles at close ranges to accuracies better than one milliradian standard deviation. As an alternative to the correlation tracker, an adaptive extended Kalman filter is developed in this research. This Kalman filter does not process correlation tracker outputs but uses FLIR outputs directly.

Research Outline

In 1978, Captain Daniel Mercier completed an initial thesis which demonstrated the feasibility of using an extended Kalman filter that uses direct FLIR data for target position estimation (Ref 5). The target he studied was at long ranges, and could be modelled as a point source of infrared (IR) radiation. Further, the target could be well described dynamically by a zero-mean first order Gauss-Markov process for position in each direction of the two planar axes in the FLIR field of view. The target motion model had a small RMS (root mean square) value resulting in small excursions for each sample period of the FLIR sensor. Mercier's analysis consisted of a "truth model" to represent real world target motion and disturbance environment, and a filter (tracker) to estimate the target motion from measurements generated from the truth model (see Appendix A for details).

Mercier's truth model and filter were the starting point for the research presented in this paper. Initially, the robustness of Mercier's filter was tested by running computer simulations with truth model and filter parameter mismatches. Next, major truth model changes were made to accommodate the new close-range, air-to-air missile target scenario, and Mercier's filter was again tested against the new truth model in computer simulations. The results of

the two sets of simulations indicated the areas of appropriate study for filter improvement. This list of improvement areas was prioritized according to their effect on filter performance. Subsequently, filter changes were made in a logical sequence, and computer simulations were run to evaluate new performance capabilities. The result is a filter which could potentially meet all AFWL requirements.

Having described the research goals in general terms, certain assumptions are required to further define and limit the problem.

Assumptions

FLIR. The FLIR outputs are instantaneous samples of an array of infrared detectors which are mechanically scanned by a many faceted mirror through a restricted field of view. Each minute infrared detector emits electrical current proportional to the intensity of the infrared photons entering the face of the detector. A single digitized output represents a real time, electronic spatial average of n horizontally scanned detectors. The serial digitized data can either be stored or displayed on a cathode ray tube (CRT); each picture element of the CRT is called a pixel. The horizontal scanning of the detectors proceeds vertically through the FLIR field of view resulting in an array (frame) of pixels which is analogous in size (about 500 by 400 elements) and appearance to a normal TV picture. A new frame of pixels is generated every thirtieth of a second (30 Hz frame rate). For this study, an 8-by-8 array of pixels out of each large frame of pixels constitutes a single measurement array (often called a tracking window). Restricting the size of the measurement array is primarily dictated by computational and storage limitations and partially justified by fast measurement rates (to be analyzed later). (Ref 5:4-5)

The digitized FLIR data has notable biases in the alternating rows of data. The elements of two rows (called rows A & B) of pixels are generated alternately during each horizontal scan. With a clear field of view (i.e. FLIR pointed at blue sky) the average of an A and B row of outputs will differ by one or two step counts. This quirk in the data can be easily compensated so it is not modelled in the study simulation program. However, this data discrepancy must be accommodated when analyzing real digital data for spatial and temporal correlations.

Target. The target for current AFWL tests and this research is an air-to-air missile. At close ranges, a missile will appear to have its own distinct geometrical shape. Previously, Mercier modelled the long range target as a point source, yielding a bivariate Gaussian intensity profile with circular, equal intensity contours on the FLIR image plane. For close range missiles, an extension to this model to allow the bivariate Gaussian model to have elliptical, equal intensity contours with specified angular orientation is used. Real FLIR data supports the use of this model to approximate closely missile shape in the image plane (FLIR focal plane).

Air-to-air missiles can exhibit high velocities as well as high-g turns - a very different target motion from that which Mercier studied. This will require substantial alterations to the assumed model of target dynamics.

To track an air-to-air missile, its image must be kept within the 8-by-8 field of view. For a benign target (very slow moving), an estimate of only position offset variables would be needed to provide to the controller as the intensity pattern moves less than half the field of view in one sample period. However, the speed of a missile enables the intensity pattern to leave the field of view in one sample period. As a minimum, target velocity is needed in addition

to current position offset to predict subsequent target position.

Intensity Pattern. The intensity pattern of the missile's projection on the FLIR focal plane is well modelled by elliptical, equal intensity contours (Ref 14:15). These elliptical contours can be described mathematically via a bivariate Gaussian intensity function.

For this model to be useful, several parameters must be known: the size of the major and minor axes and the orientation of principal axes in the image plane. Figure 1 illustrates an intensity function in FLIR coordinates.

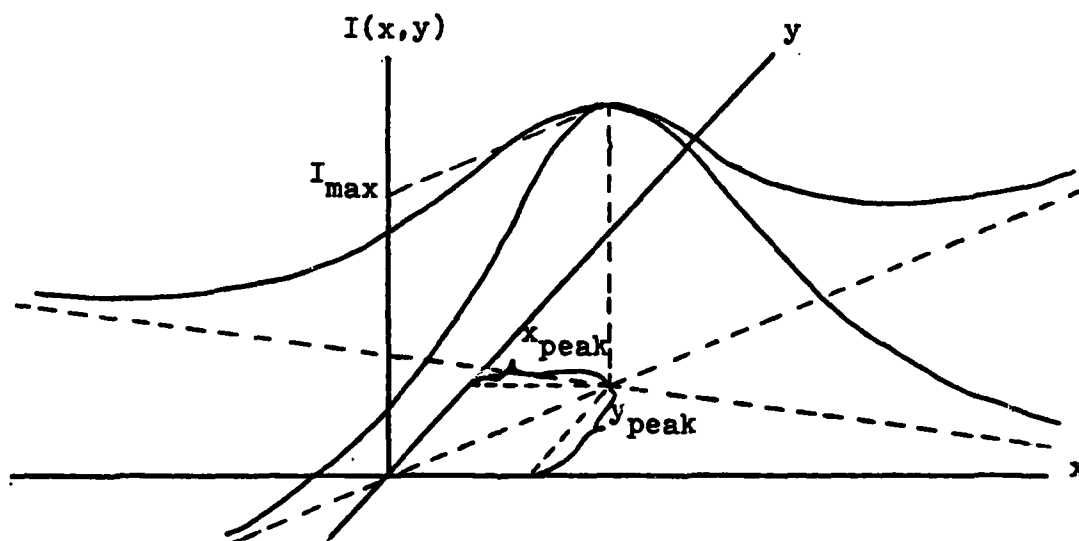


Figure 1. Gaussian Bivariate Distribution Intensity Model, $I(x,y)$

The equation for this intensity function is

$$I(x,y) = I_{\max} \exp \left\{ \frac{-1}{2(1-r)^2} \left[\frac{(x-x_{\text{peak}})^2}{\sigma_x^2} + \frac{(y-y_{\text{peak}})^2}{\sigma_y^2} - 2r \frac{(x-x_{\text{peak}})(y-y_{\text{peak}})}{\sigma_x \sigma_y} \right] \right\} \quad (1)$$

where I_{\max} = the maximum target intensity

x, y = the coordinates of any point in the FLIR focal plane

$x_{\text{peak}}, y_{\text{peak}}$ = coordinates locating peak intensity,
 I_{\max}

r = Image dispersion correlation

σ_x, σ_y = Image dispersions in the x and y directions respectively

The intensity function would appear as elliptical, equal intensity contours in the FLIR as shown in Figure 2.

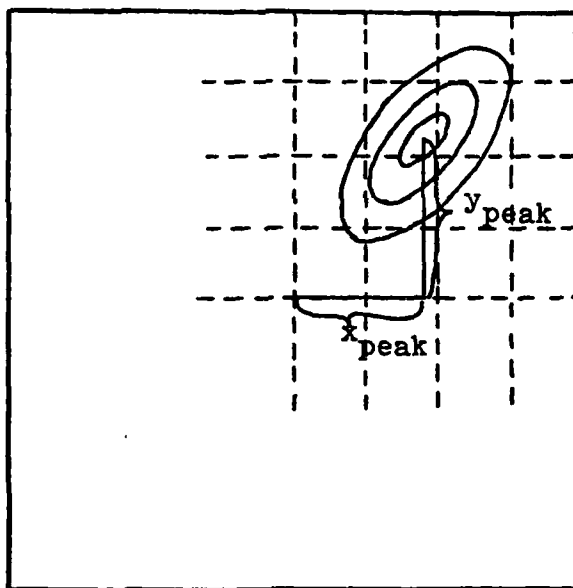


Figure 2. Target Image in FLIR Field of View

The apparent target intensity function location in the FLIR field of view consists of several components. Bore-sight error, FLIR system vibrations and others are assumed to be negligible so that the intensity function location can be centered by:

$$\begin{bmatrix} x_{\text{peak}}(t) \\ y_{\text{peak}}(t) \end{bmatrix} = \begin{bmatrix} x_D(t) + x_A(t) \\ y_D(t) + y_A(t) \end{bmatrix} \quad (2)$$

where x_D, y_D = position offsets due to target dynamics
 x_A, y_A = position offsets due to atmospheric jitter

The separation of x_D from x_A and y_D from y_A is desirable for the target tracker, a capability not possible with correlation trackers.

Background Noise. The image background noise which, along with FLIR sensor noise, contaminate the FLIR measurements, is modelled as a spatially and temporally correlated, Gaussian process. Various physical backgrounds result in FLIR images with differing spatial and temporal correlation characteristics. Real data analysis and AFWL experience has been used to determine appropriate correlation coefficients for the spatial and temporal correlations.

Closed Loop. This is a closed loop tracking system. The laser pointing system is assumed to be perfect. That is, the system can point exactly where the tracker commands it within each sample time. This implies that settling time of the pointing system is less than the data sample period, the time between discrete samples of the FLIR output.

Previous Investigation

The truth model and filter derived by Mercier will serve as a beginning for this research. The truth model target dynamics were modelled as independent, first order Gauss-Markov processes in the two FLIR component directions, written mathematically as:

$$\dot{x}_D(t) = (-1/\tau_D)x_D(t) + w_1(t) \quad (3)$$

$$\dot{y}_D(t) = (-1/\tau_D)y_D(t) + w_2(t) \quad (4)$$

where w_1 and w_2 are independent white Gaussian noises and τ_D is the dynamics correlation time (refer to Appendix A for a more detailed description of Mercier's work) (Ref 5: 9-10).

The truth model atmospheric disturbances were modelled as third order Gauss-Markov processes based on studies by The Analytical Sciences Corporation (TASC) (Ref 6,7). As part of this study, TASC generated a model for x_A , the atmospheric jitter, in equation (2) as the output of a system with a frequency domain transfer function of

$$\frac{x_A}{w_3} = \frac{K(14.14)(659.5)^2}{(s + 14.14)(s + 659.5)^2} \quad (5)$$

as driven by a unit strength white Gaussian noise w_3 . K in equation (5) is adjusted to establish the current root mean square (RMS) value of x_A . y_A is similarly modelled. (See Appendix A for conversion of equation (5) to Jordan canonical form.)

These models resulted in an eight state truth model with one dynamic state and three atmospheric states for each direction. The truth model state vector is propagated, after conversion to equivalent discrete time form, by the following equation:

$$\underline{x}_T(t_{i+1}) = \underline{\phi}_T \underline{x}_T(t_i) + \underline{w}_d(t_i) \quad (6)$$

where $\underline{x}_T(t_{i+1})$, $\underline{x}_T(t_i)$ = state vector at times t_{i+1} and t_i , respectively

$\underline{w}_d(t_i)$ = discrete noise term

$\underline{\phi}_T$ = constant state transition matrix (Ref 5:15)

The truth model target intensity function is converted into measurements by the following relationship:

$$z_{kl}(t_i) = \frac{1}{A_p} \iint_{\substack{\text{i-jth} \\ \text{pixel}}} I_{\max} \exp \left\{ -\frac{1}{2\sigma_g^2} \left[(x-x_{\text{peak}}(t_i))^2 + (y-y_{\text{peak}}(t_i))^2 \right] \right\} dx dy + n_{kl}(t_i) + f_{kl}(t_i) \quad (7)$$

where $z_{kl}(t_i)$ = output of the i-jth pixel at time t_i

A_p = area of a pixel

x, y = coordinates of any point in the pixel

$x_{\text{peak}}(t_i), y_{\text{peak}}(t_i)$ = coordinates of the center of the apparent intensity distribution with respect to the center of the 8-by-8 FLIR field of view at time t_i

I_{\max} = maximum target intensity

σ_g = dispersion of the Gaussian intensity function

$n_{kl}(t_i)$ = background noise term for the i-jth pixel

$f_{kl}(t_i)$ = FLIR noise term for the i-jth pixel

(Ref 4:17)

The filter derived by Mercier had four states, one state for target dynamics and one state for atmospheric jitter in each direction. Both target dynamics and atmospheric jitter were modelled as first order Gauss-Markov processes driven by white Gaussian noise as follows

$$\dot{x}_D(t) = -\frac{1}{\tau_D} x_D(t) + w_1(t) \quad (8)$$

$$\dot{x}_A(t) = -\frac{1}{\tau_A} x_A(t) + w_2(t) \quad (9)$$

where τ_D = correlation time of target dynamics

$w_1(t), w_2(t)$ = independent white Gaussian noise processes

τ_A = correlation time of atmospheric jitter, 1/14.14

Analogous equations were used for $y_D(t)$ and $y_A(t)$ (Ref 5:19). Equation (9) is a reasonable approximation of equation (5) in view of the discrepancy of the break frequencies and the greater importance of the lower frequency characteristics. After deriving the state transition matrix, $\Phi_F(t, t_i)$, and $Q_{Fd}(t_i)$, the discretized noise strength matrix, the filter states and covariance matrix are propagated using standard extended Kalman filter forms (see Appendix A). Mercier used the following initial conditions for his filter:

$$\underline{x}_F(t_0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad \underline{P}(t_0) = \begin{bmatrix} \sigma_{D_0}^2 & 0 & 0 & 0 \\ 0 & \sigma_{D_0}^2 & 0 & 0 \\ 0 & 0 & \sigma_{A_0}^2 & 0 \\ 0 & 0 & 0 & \sigma_{A_0}^2 \end{bmatrix} \quad (10)$$

where $\underline{P}(t_0)$ = initial covariance estimate

$\sigma_{D_0}^2$ = initial estimate of dynamics variance

$\sigma_{A_0}^2$ = initial estimate of variance of atmospheric jitter

The measurement update of Mercier's extended Kalman filter is more unique and noteworthy. First of all, for the design of the filter the measurements (64 pixel outputs) are assumed to be a nonlinear function of the filter states plus noise as follows:

$$\underline{z}(t_i) = \underline{h}(\underline{x}_F(t_i), t_i) + \underline{v}(t_i) \quad (11)$$

where \underline{h} represents the effect of the intensity spread function given in (1). This equation results in the following

measurement update equations in standard extended Kalman filter form:

$$\underline{K}(t_i) = \underline{P}(t_i^-) \underline{H}^T(t_i) [\underline{H}(t_i) \underline{P}(t_i^-) \underline{H}^T(t_i) + \underline{R}(t_i)]^{-1} \quad (12)$$

$$\hat{\underline{x}}(t_i^+) = \hat{\underline{x}}(t_i^-) + \underline{K}(t_i) [\underline{z}(t_i) - \underline{h}(\hat{\underline{x}}(t_i^-), t_i)] \quad (13)$$

$$\underline{P}(t_i^+) = \underline{P}(t_i^-) - \underline{K}(t_i) \underline{H}(t_i) \underline{P}(t_i^-) \quad (14)$$

where
$$\underline{H}(t_i) = \left. \frac{\partial \underline{h}(\underline{x}, t)}{\partial \underline{x}} \right|_{\underline{x} = \hat{\underline{x}}(t_i^-)} \quad (15)$$

However, equation (12) would require the inversion of a 64-by-64 matrix. To avoid this, the inverse covariance form of the update cycle was used (Ref 8:238-241) which requires only two 4-by-4 matrix inversions:

$$\underline{P}(t_i^+) = [\underline{H}^T(t_i) \underline{R}^{-1}(t_i) \underline{H}(t_i) + \underline{P}^{-1}(t_i^-)]^{-1} \quad (16)$$

$$\hat{\underline{x}}(t_i^+) = \underline{x}(t_i^-) + \underline{P}(t_i^+) \underline{H}^T(t_i) \underline{R}^{-1}(t_i) [\underline{z}(t_i) - \underline{h}(\hat{\underline{x}}(t_i^-), t_i)] \quad (17)$$

Truth Model Changes

Capt. Mercier's truth model was modified in several ways to accommodate this research. One change was incorporated to simulate an air-to-air missile flight path. This flight path was programmed by adding a deterministic input to the state equation.

$$\underline{x}(t_{i+1}) = \underline{x}(t_i) + \underline{B}u(t_i) \quad (18)$$

where \underline{B} is a constant input matrix and $u(t_i)$ is an equivalent discrete time input function depicting the trajectory.

To simulate the measurements taken from the FLIR image plane, the truth model was modified to use position and velocity of the missile to orient an ellipsoid in 3D space. Assuming the missile velocity vector is colinear with the major axis of the ellipsoid, the major axis of the ellipsoid is projected onto the FLIR image plane. This projection and the minor axis of the ellipsoid are then used as semi-major and semi-minor axes of the elliptical equal intensity contours.

Mercier modelled the disturbances in the measurements due to the background and FLIR (represented by $n_{kl}(t)$ and $f_{kl}(t)$ in equation (7)) as spatially uncorrelated, white Gaussian noises. However, most FLIR measurements exhibit both spatial and temporal relationships depending on the background being observed. The spatial background is easily handled by generating spatially correlated noise using a measurement noise strength matrix with appropriate cross correlation terms (off-diagonal) filled. The modelling of the temporal relationship of these noises is much more complicated. The model chosen was to represent the time history of the noise for each pixel as the output of a first order lag system (yielding exponentially time-correlated output) driven by spatially correlated noise. Thus, 64 new states were added to the truth model to accomplish the simulation of this effect.

II. Truth Model Development

Introduction

The truth model, a mathematical representation of the real world system to be evaluated, is developed with those characteristics needed to model the close range air-to-air missile problem accurately. Starting with the previous truth model as described in Chapter I and Appendix A, the truth model simulation package is supplemented with several large modelling changes needed to account for the change in the target scenarios. The changes incorporated are the addition of spatial and temporal correlations to the FLIR background noise, the portrayal of the target as an intensity pattern with elliptical constant intensity contours, the motion of air-to-air missile at close ranges, and the effects of changing missile image size and shape with varying range in a fixed field of view.

As an alternative to the development of a truth model, the use of an AFWL-supplied routine called REFERENCING is discussed. This routine is capable of reading actual site test data from tape. This software package has several capabilities and has considerable potential as a future analysis tool.

Spatially Correlated Background Noise

The modelling of spatial correlations in the background noise is conceptually easy but laborious in implementation. Before the mechanics for creating spatially correlated background noise are presented, the need for this modelling is discussed.

The existence of spatial correlations in the background of the FLIR data has been observed by AFWL personnel and can be seen in real data analysis. Captain Lewantowicz of the AFWL has recommended spatial correlations for distances up to 40 microradians (μrad) based on his experience (Ref 9).

Forty microradians is approximately equivalent to two pixels in the scale used in this research.

Evaluation of real data, which is explained in detail in Appendix B along with tables and results, further demonstrates that spatial correlations exist in the background of the FLIR field of view. The data analyzed was actual digital FLIR values taken while tracking an approaching air-to-air missile. The background for the test started as clear sky and changed to ground clutter when the missile's motion carried it below the horizon. The data values analyzed were the sum of one, four, or sixteen pixels (i.e. from individual pixels, 2-by-2 arrays, or 4-by-4 arrays) as needed to maintain the target image in the limited field of the data matrix. The results of the analysis, which are found in Tables B-I through B-IV of Appendix B, are summarized here. The correlation coefficients between adjacent pixels (first neighbor) in the horizontal direction averaged 0.325 (the first eight entries of Table B-I) for values which represented a single pixel's output. The horizontal coefficients of values representing the sum of four or sixteen pixels (the last six entries of Table B-I) averaged 0.25 - slightly higher than expected because of the large distance which this spatial correlation represents. Such pixel summations were done near the end of the missile's flight when the missile had gone below the local horizon. Thus, the higher than expected correlations of the multiple pixel values may have been due to a change of physical background, a background with more structure.

The computation of vertical, first neighbor coefficients was complicated by the bias in the average of alternating rows (called A and B) of data as originally discussed in Chapter I. In fact, as can be seen in the first six entries of Table B-III, the correlation coefficients computed when comparing A and B values are primarily negative. When A and B values are added together and then correlated,

as occurs in the last five entries of Table B-III, the coefficients were positive and average 0.355.

Similarly, correlation coefficients between a pixel and the pixel two away (considering any three pixels in a row, the first and third are compared) in the vertical and horizontal directions are computed from real data (Appendix B, Tables B-II and B-IV). Horizontally, the data was unexpectedly uncorrelated except when four or more pixels were added. This may be due in part to the character of the background which was clear sky when single pixel values were used, as in the first eight entries of Table B-II. In the vertical direction when four and sixteen pixel sums, which adds together A and B values, were correlated, the coefficients (the last five entries of Table B-IV) averaged 0.334.

Thus, the real data analysis of single pixels substantiated the existence of the spatial nature of the background of FLIR data, but the correlations were not found to be as extensive as expected due probably to the sky background. Other than the last segment of this test run, all available data sources had clear sky backgrounds. Consequently, the spatial contents of other backgrounds were not studied.

Another effect possibly contributing to the horizontal correlations was the horizontal sweeping of the detectors. This effect was not noticeable in the data analysis because the A and B row effects did not allow a good comparison of vertical and horizontal coefficients. More methodical data analysis of known backgrounds is needed to establish the horizontal contribution of this effect and to establish the range of reasonable correlations.

The generation of spatially correlated white Gaussian noises is accomplished by allowing nonzero cross correlations between the measurement noises $v_j(t_i)$ in the following equation.

$$\underline{z}(t_i) = \underline{h}(x(t_i), t_i) + \underline{v}(t_i) \quad (19)$$

The components of $\underline{v}(t_i)$, which is of dimension 64, are no longer independent as in the original truth model but are spatially correlated. The noise covariance matrix is then no longer diagonal but has the general form

$$E\{\underline{v}(t_i)\underline{v}^T(t_j)\} = \underline{R} \delta_{ij}$$

$$\underline{R} = \begin{bmatrix} \sigma_1^2 & r_{1,2}\sigma_1\sigma_2 & r_{1,3}\sigma_1\sigma_3 & \cdots & r_{1,64}\sigma_1\sigma_{64} \\ r_{2,1}\sigma_2\sigma_1 & \sigma_2^2 & r_{2,3}\sigma_2\sigma_3 & \cdots & r_{2,64}\sigma_2\sigma_{64} \\ r_{3,1}\sigma_3\sigma_1 & r_{3,2}\sigma_3\sigma_2 & \sigma_3^2 & \cdots & r_{3,64}\sigma_3\sigma_{64} \\ \cdots & \cdots & \cdots & \ddots & \cdots \\ r_{64,1}\sigma_{64}\sigma_1 & r_{64,2}\sigma_{64}\sigma_2 & r_{64,3}\sigma_{64}\sigma_3 & \cdots & \sigma_{64}^2 \end{bmatrix} \quad (20)$$

where \underline{R} = 64-by-64 measurement noise covariance matrix

r_{kl} = correlation coefficient relating noise components v_k and v_l at a given time t_i

\underline{v} is assumed stationary so \underline{R} is constant for a given background; it would be changed to reflect a significant change in the physical background. The correlation coefficient is defined statistically for zero mean random variables as:

$$r_{k,l} = \frac{E\{v_k(t_i)v_l(t_i)\}}{\sqrt{E\{v_k^2(t_i)\}E\{v_l^2(t_i)\}}} = \frac{E\{v_k(t_i)v_l(t_i)\}}{\sigma_k\sigma_l} \quad (21)$$

From the above equation, it can be seen that $r_{kl} = r_{lk}$ since σ_k , σ_l , v_k , and v_l are all scalars and that $r_{kk} = 1$. These facts and the fact that the variances of the background noises are assumed to have the same strength ($\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_{64}^2 = \sigma_R^2$) yields a simplified noise covariance matrix:

$$\underline{R} = \sigma_v^2 \begin{bmatrix} 1 & r_{1,2} & r_{1,3} & \cdots & r_{1,64} \\ r_{1,2} & 1 & r_{2,3} & \cdots & r_{2,64} \\ r_{1,3} & r_{2,3} & 1 & \cdots & r_{3,64} \\ \cdots & \cdots & \cdots & \ddots & \cdots \\ r_{1,64} & r_{2,64} & r_{3,64} & \cdots & 1 \end{bmatrix} \quad (22)$$

At this point the relationship between models for correlated and uncorrelated spatial backgrounds is most readily understood. For uncorrelated Gaussian noises, $r_{k,l}$ in (22) equal zero. However, if the noises are spatially correlated, the noise in pixel k is correlated due to its physical proximity to pixel l , then some or all of $r_{k,l}$ in (22) are nonzero.

Having explained what is meant by spatially correlated noise, the next step is to determine the amount of spatial correlation that exists in actual data and then to reflect this in \underline{R} . AFWL inputs and real data analysis have shown that the physical limit of such correlations are one or two pixels distance (first and second neighbors). If the pixels in the 8-by-8 measurement array are numbered 1 through 64 starting in the upper left and progressing left to right, row by row (see Figure 3), and if the pixel noises are similarly numbered, then the correlation coefficients can be assigned appropriate values. Shown in Figure 4 are the nonzero correlation coefficients for pixel k for correlations limited to its first neighbor. The number in the upper right hand corner is the pixel number with respect to pixel k . The coefficient in each box is that which relates the noise in that box with the noise in pixel k . For first neighbor correlations all coefficients are zero except those relating pixel k to its eight immediate neighbors. The real data analysis indicated that the coefficients were symmetric vertically and horizontally; the four diagonal

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Figure 3. Pixel Numbering Scheme

$k-9$ $r_{k,k-9}$	$k-8$ $r_{k,k-8}$	$k-7$ $r_{k,k-7}$
$k-1$ $r_{k,k-1}$	k $r_{k,k=1}$	$k+1$ $r_{k,k+1}$
$k+7$ $r_{k,k+7}$	$k+8$ $r_{k,k+8}$	$k+9$ $r_{k,k+9}$

Figure 4. First Neighbor Correlation Coefficients

coefficients were set equal to each other, as were the four neighbors above, below, and to the sides. Thus, $r_{k,k-9}$, $r_{k,k-7}$, $r_{k,k+7}$, and $r_{k,k+9}$, shown in Figure 4, were set equal to .2545 for the first neighbor exponential curve fit to be discussed subsequently, and $r_{k,k-8}$, $r_{k,k-1}$, $r_{k,k+1}$, and $r_{k,k+8}$ were set equal to .3800 for the first neighbor exponential curve fit. Making similar observations of the correlation coefficients for all the pixels, the \underline{R} matrix can be computed.

Similarly, if the spatial correlations are to be modelled as extending to first and second neighbors, then the coefficients relating a pixel's noise to that of its 24 closest neighbors are nonzero as shown in Figure 5, and all others are zero. Making appropriate selections for all pixels allows \underline{R} to be computed, but this time it has considerably more nonzero entries. Actual implementation in the computer simulation software (see Appendix G for computer code listings) of the \underline{R} formulation is simplified by the symmetry of the coefficients about the main diagonal and by the use of recursive loops.

$k-18$ $r_{k,k-18}$	$k-17$ $r_{k,k-17}$	$k-16$ $r_{k,k-16}$	$k-15$ $r_{k,k-15}$	$k-14$ $r_{k,k-14}$
$k-10$ $r_{k,k-10}$	$k-9$ $r_{k,k-9}$	$k-8$ $r_{k,k-8}$	$k-7$ $r_{k,k-7}$	$k-6$ $r_{k,k-6}$
$k-2$ $r_{k,k-2}$	$k-1$ $r_{k,k-1}$	k $r_{k,k=1}$	$k+1$ $r_{k,k+1}$	$k+2$ $r_{k,k+2}$
$k+6$ $r_{k,k+6}$	$k+7$ $r_{k,k+7}$	$k+8$ $r_{k,k+8}$	$k+9$ $r_{k,k+9}$	$k+10$ $r_{k,k+10}$
$k+14$ $r_{k,k+14}$	$k+15$ $r_{k,k+15}$	$k+16$ $r_{k,k+16}$	$k+17$ $r_{k,k+17}$	$k+18$ $r_{k,k+18}$

Figure 5. First and Second Neighbor Correlations

Once \underline{R} is known, specific spatially correlated realizations of \underline{y} can be generated by using a Cholesky square

root decomposition of \underline{R} and a Gaussian noise generator with unit variance. Specifically, $\underline{v}(t_i)$ can be generated as

$$\underline{v}(t_i) = \sqrt{\underline{R}} \underline{w}(t_i) \quad (23)$$

where $\underline{w}(t_i)$ = 64 dimension vector of independent white noises of unit variance generated by independent calls to a Gaussian noise routine, i.e., $E\{\underline{w}(t_i)\underline{w}(t_j)\} = \underline{I} \delta_{ij}$.

and $\sqrt{\underline{R}}$ is the Cholesky square root of \underline{R} as discussed in Chapter 7 of Reference 10. The Cholesky decomposition produces a lower triangular matrix such that

$$\sqrt{\underline{R}} \sqrt{\underline{R}}^T = \underline{R} \quad (24)$$

The covariance of \underline{v} as given in (20) is preserved with the use of this decomposition as follows:

$$\begin{aligned} E\{\underline{v}(t_i)\underline{v}^T(t_j)\} &= E\{\sqrt{\underline{R}} \underline{w}(t_i)\underline{w}^T(t_j) \sqrt{\underline{R}}^T\} \\ &= \sqrt{\underline{R}} E\{\underline{w}(t_i)\underline{w}^T(t_j)\} \sqrt{\underline{R}}^T = \sqrt{\underline{R}} \underline{I} \sqrt{\underline{R}}^T \delta_{ij} \\ &= \underline{R} \delta_{ij} \end{aligned} \quad (25)$$

The final step in the spatial correlation model is establishing the appropriate values for the correlations. The early trial runs and real data analysis provided considerable insight into the formulation of a good model of the correlation curve. Initial runs were attempted with high (r values ranging from .95 to .8) first and second neighbor correlations in \underline{R} and zero correlations elsewhere, which is physically an unrealistic trend for the spatial correlations to follow. This resulted in \underline{R} being numerically ill-conditioned.

The Cholesky decomposition of \underline{R} could not be computed as \underline{R} is required to have non-negative eigenvalues (10:7-5) as computed on a fixed word length computer. The Cholesky decomposition will theoretically work for any symmetric, positive semidefinite matrix given an infinite word length computer. Because all sensors have some noise, \underline{R} should be positive definite. Thus, special care must be taken in choosing correlation coefficients that make sense physically so that \underline{R} has numerically computable positive eigenvalues. The real data and the practical experience of Dr. P. S. Maybeck suggested that the correlation could be modelled adequately as a decaying exponential. The real data analysis indicated first neighbor correlations of about .3 to .4 and second neighbor correlations of essentially zero. Remembering that autocorrelations yield unity correlation coefficients, these three points (1,0; .35,1; .0,2) suggest an exponential decay or other nonlinear curve fit. The exponential decay model with correlation distance of one pixel was chosen because it fit the data for the first neighbor and was essentially zero for the second neighbor and beyond. The exponential model also prevented \underline{R} from being ill-conditioned. There was an upper limit found for the first neighbor correlations and for the first and second neighbor correlations, above which \underline{R} became ill-conditioned. For first neighbor correlations this value was about .4, and for first and second neighbor correlations it was about .5 for the closest neighbor and .3 for the second neighbor.

In summary, the following algorithm can be used to generate a vector, $\underline{v}(t_i)$, of spatially correlated noises:

- (1) Decide on the appropriate curve fit for the spatial correlations and compute \underline{R} .

- (2) Compute $\sqrt{\underline{R}}$.

- (3) Generate $\underline{w}(t_i)$, the vector of independent white noises of unit variance.

- (4) Compute $\underline{v}(t_i)$ by premultiplying $\underline{w}(t_i)$ by $\sqrt{\underline{R}}$.

Elliptical Intensity Contour Target

The introduction outlined in a general fashion the desirability of using elliptical constant intensity contours to model an air-to-air missile image. There are several good reasons for doing so. Studying frames of data from AFWL indicated that for any particular intensity level the pattern was approximately elliptical and in some cases circular. This seems to bear out that the atmosphere and optics tend to smear a well defined image and attenuate the edges gradually. The diffraction pattern produced by the image closely approaches a Gaussian intensity function (Ref 14:15). The use of a bivariate Gaussian intensity function to simulate this effect maintains the essence of the observed image properties and mathematical ease of handling.

Capt. Mercier assumed the object to be far enough away so as to appear as a point source of radiation. Due to the physics of wave propagation and optics, this yields an intensity pattern on the FLIR image plane well approximated by a bivariate Gaussian function with circular equal intensity contours. To simulate closer range targets the point source assumption was clearly inadequate. This, along with observed data, motivated the elliptical shape with specified orientation in the image plane used in this research.

The intensity function used is

$$I(x,y) = I_{\max} \exp\left\{-\frac{1}{2}[(x-x_{\text{peak}})(y-y_{\text{peak}})][P]^{-1}[(x-x_{\text{peak}})(y-y_{\text{peak}})]^T\right\} \quad (26)$$

where the variables are shown in Figure 6; x_{peak} and y_{peak} locate the peak of the intensity function, and σ_v and σ_{pv} are the eigenvalues of the P matrix in the equation above. To quantify the image, a pixel dimension of 20 μ rads is used. Therefore, an 8 x 8 FOV will be 160 μ rads wide in

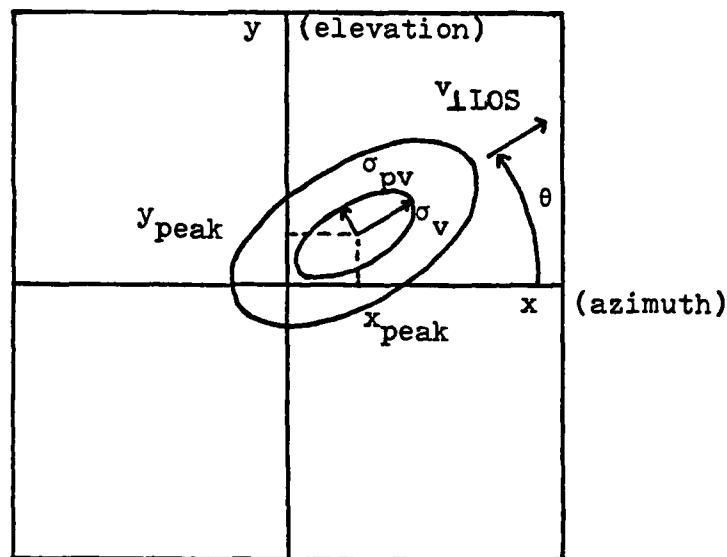


Figure 6. Image Intensity Characteristics

azimuth and 160 μ rad wide in elevation. Because of the small FOV, angular displacement of the target from the center of the FOV is closely approximated by linear displacement on the FLIR image plane. Also, angular velocity closely approximates image plane linear velocity. Since the distinction between angular displacement and linear displacement is negligible, angular measurements were used in this research.

The missile image was produced in several steps by the truth model. From the simulated three-dimensional inertial position and velocity of the missile, the azimuth velocity (x velocity in Figure 6), elevation velocity (y velocity in Figure 6) and speed (magnitude of velocity vector in 3-D space) were computed in pixels/sec. The azimuth velocity and elevation velocity then define the missile velocity component perpendicular to the line of sight (LOS). The ratio of the velocity perpendicular to the LOS and speed is the cosine of the out of plane angle γ . Figure 7 shows the geometry involved.

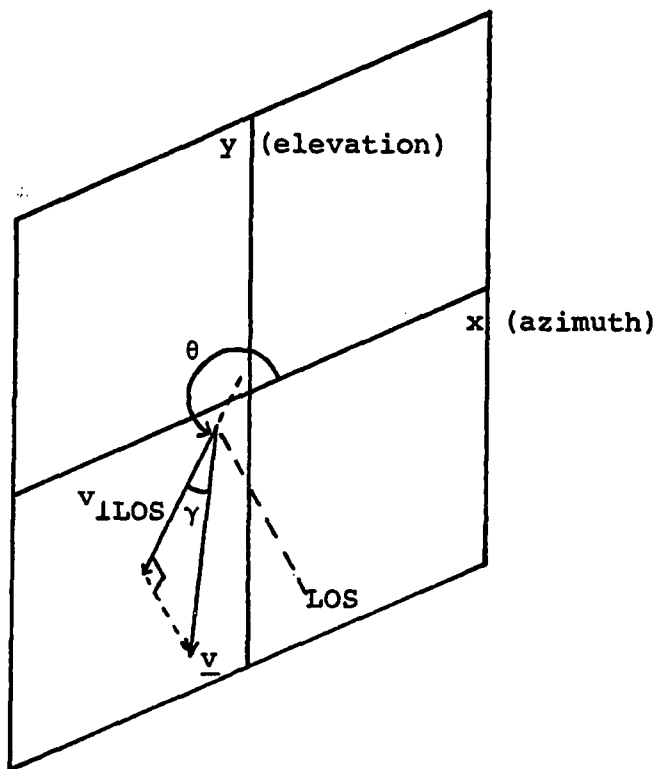


Figure 7. Image Projection

γ = out of plane angle

θ = orientation angle in image plane

$$\cos\theta = \frac{\text{azimuth velocity}}{|v_{\perp\text{LOS}}|} \quad (27)$$

$$\sin\theta = \frac{\text{elevation velocity}}{|v_{\perp\text{LOS}}|} \quad (28)$$

The missile is assumed to be oriented along its velocity vector. This is not strictly due to nonzero angle of attack and side slip angle but the error introduced was thought to be insignificant for the purposes of this research.

With $\cos\theta$ determined, the semimajor axis of the three dimensional missile is projected onto the image plane. This projected length becomes σ_v and the radius of the circular missile IR cross-section is retained as the semiminor axis magnitude, σ_{pv} , in the image plane. With these parameters established, the intensity at any point in the image plane can be computed. This computation is performed in the principal axis coordinate system of the image ellipse. The intensity function then becomes

$$I(x,y) = I_{\max} \exp \left\{ -\frac{1}{2} [\Delta x_v \Delta y_v] \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_{pv}^2 \end{bmatrix}^{-1} [\Delta x_v \Delta y_v]^T \right\} \quad (29)$$

$$\text{where } \Delta x_v = (x - x_{\text{peak}}) \cos \theta + (y - y_{\text{peak}}) \sin \theta \quad (30)$$

$$\Delta y_v = (y - y_{\text{peak}}) \cos \theta - (x - x_{\text{peak}}) \sin \theta \quad (31)$$

where the rotation angle θ between (x,y) and principal axis coordinates is given in Figures 6 and 7.

The average intensity for any pixel, as measured by the FLIR, is the integral of the intensity function over the pixel area divided by the pixel area. To approximate this integral, the intensity was averaged over sixteen equally spaced points inside the pixel (i.e. at the centers of a 4-by-4 array of squares in each pixel).

To complete the average intensity measurement simulation, noise due to various FLIR and background sources was added to each pixel. Finally, the intensity function, including noise terms, for one of the 64 pixels is

$$z_{lm}(t_i) = \frac{1}{16} \sum_{k=1}^4 \sum_{j=1}^4 I_{\max} \left\{ -\frac{1}{2} [\Delta x_{vk} \Delta y_{vj}] \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_{pv}^2 \end{bmatrix}^{-1} [\Delta x_{vk} \Delta y_{vj}]^T \right\} + n_{lm}(t_i) \quad (32)$$

where the noise term $n_{lm}(t_i)$ can be modelled several different ways. Three models were used in this research.

- 1) Spatially and temporally white Gaussian discrete-time noise where

$$E[\underline{n}(t_i)] = \underline{0} \quad (33)$$

$$E[\underline{n}(t_i)\underline{n}^T(t_i)] = \underline{P} \quad (\underline{P} \text{ a diagonal matrix}) \quad (34)$$

$$E[\underline{n}(t_i)\underline{n}^T(t_j)] = \underline{0} \quad i \neq j \quad (35)$$

- 2) Spatially correlated, temporally white Gaussian noise where

$$E[\underline{n}(t_i)] = \underline{0} \quad (36)$$

$$E[\underline{n}(t_i)\underline{n}^T(t_i)] = \underline{P} \quad (\underline{P} \text{ not a diagonal matrix}) \quad (37)$$

$$E[\underline{n}(t_i)\underline{n}^T(t_j)] = \underline{0} \quad t_i \neq t_j \quad (38)$$

- 3) Spatially and temporally correlated Gaussian noise where

$$E[\underline{n}(t_i)] = \underline{0} \quad (39)$$

$$E[\underline{n}(t_i)\underline{n}^T(t_i)] = \underline{P} \quad (\underline{P} \text{ not a diagonal matrix}) \quad (40)$$

$$E[\underline{n}(t_i)\underline{n}^T(t_j)] = \underline{P} e^{-|t_i - t_j|/\tau_A} \quad (41)$$

$$\tau_A = \text{correlation time of noise}$$

With zero mean noise some pixels far from the target could have negative intensity values. This was previously prevented by searching out the minimum value of the intensity pattern and setting this pixel value arbitrarily to .1. The other pixels were adjusted by the same amount to maintain

the correct image (Ref 5:90). This scaling was retained for this research. The image produced by the collection of 64 $z_{1m}(t_i)$ is presented to the tracker as the measured data.

Target Dynamics Model

The continuous time dynamic model for the air-to-air missile used in this truth model simulation was

$$\dot{x}_1 = \dot{\alpha}(t) \quad (42)$$

$$\dot{x}_2 = \dot{\beta}(t) \quad (43)$$

with
$$\underline{u}(t) = [\dot{\alpha}(t) \ \dot{\beta}(t)]^T$$

In vector form, this equation becomes

$$\dot{\underline{x}}_D(t) = \underline{u}(t) \quad (44)$$

where $\underline{u}(t)$ is a specified azimuth and elevation velocity time function used as the input to program a particular trajectory. Equation (44) has the general solution

$$\underline{x}_D(t) = \underline{\phi}_D(t, t_0) \underline{x}(t_0) + \int_{t_0}^t \underline{\phi}_D(t, \tau) \underline{B}_D(\tau) \underline{u}(\tau) d\tau \quad (45)$$

where t_0 = initial time

$\underline{\phi}_D(t, \tau)$ = state transition matrix for missile dynamics

$\underline{B}_D(\tau)$ = input matrix for dynamics

$\underline{u}(\tau)$ = input function for dynamics

The atmospheric disturbance model as described in Ref 5 is a stochastic equation of the form

$$\dot{\underline{x}}_A(t) = \underline{F}_A(t) \underline{x}_A(t) + \underline{G}_A(t) \underline{w}_A(t) \quad (46)$$

where $\underline{x}_A(t)$ = the six atmospheric noise states

$\underline{F}_A(t)$ = atmospheric plant matrix

$\underline{G}_A(t)$ = atmospheric noise input matrix

$\underline{w}_A(t)$ = vector of white Gaussian noise inputs with zero mean

and
$$E\{\underline{w}_A(t)\underline{w}_A^T(t+\tau)\} = \underline{Q}_A(t)\delta(\tau)$$

Equation (46) has the solution over a sample period starting at t_i

$$\underline{x}_A(t) = \underline{\phi}_A(t, t_i)\underline{x}_A(t_i) + \int_{t_i}^t \underline{\phi}_A(t, \tau)\underline{G}_A(\tau)\underline{w}_A(\tau)d\tau \quad (47)$$

The integral term in equation (47) requires a statistical characterization. Let

$$\underline{w}_{Ad}(t_i) = \int_{t_i}^{t_{i+1}} \underline{\phi}_A(t_{i+1}, \tau)\underline{G}_A(\tau)\underline{w}_A(\tau)d\tau \quad (48)$$

and

$$E[\underline{w}_{Ad}(t_i)] = \underline{0} \quad (49)$$

$$\begin{aligned} & E[\underline{w}_{Ad}(t_i)\underline{w}_{Ad}^T(t_i)] \\ &= \int_{t_i}^{t_{i+1}} \underline{\phi}_A(t_{i+1}, \tau)\underline{G}_A(\tau)\underline{Q}_A(\tau)\underline{G}_A^T(\tau)\underline{\phi}_A^T(t_{i+1}, \tau)d\tau \quad (50) \\ &= \underline{Q}_{Ad} \end{aligned}$$

$$E[\underline{w}_{Ad}(t_i)\underline{w}_{Ad}^T(t_j)] = \underline{0} \quad t_i \neq t_j \quad (51)$$

Note that \underline{Q}_{Ad} is not a function of time: the atmospheric disturbances are modelled as stationary processes.

For the truth model to provide $\underline{x}_A(t_i)$, realizations of $\underline{w}_{Ad}(t_i)$ are required. This can be accomplished by taking the Cholesky square root of \underline{Q}_{Ad} and postmultiplying it by a vector of unit strength white Gaussian noise. Then equation (47) can be written equivalently as

$$\underline{x}_A(t_{i+1}) = \underline{\phi}_A(t_{i+1}, t_i) \underline{x}_A(t_i) + \sqrt{\underline{Q}_{Ad}} \underline{w}_A(t_i) \quad (52)$$

where

$$E[\underline{w}_A(t_i)] = \underline{0} \quad (53)$$

$$E[\underline{w}_A(t_i) \underline{w}_A^T(t_j)] = \underline{I} \delta_{ij}$$

A more detailed development of $\sqrt{\underline{Q}_{Ad}}$ can be found in Ref 10.

Augmenting equations (45) and (52) yields the complete truth model time propagation model as

$$\underline{x}(t_{i+1}) = \underline{\phi}(t_{i+1}, t_i) \underline{x}(t_i) + \begin{bmatrix} \underline{B}_d(t_i) \\ \underline{0} \end{bmatrix} \underline{u}_d(t_i) + \begin{bmatrix} \underline{0} \\ \sqrt{\underline{Q}_{Ad}} \end{bmatrix} \underline{w}_A(t_i) \quad (54)$$

where $\underline{x}(t_i)$ = state vector of dynamic and atmospheric states

$$\underline{B}_d(t_i) = \int_{t_i}^{t_{i+1}} \underline{\phi}(t_{i+1}, \tau) \underline{B}(\tau) d\tau \quad (55)$$

and $\underline{u}_d(t_i)$ is a piecewise constant function (constant between sample times)

$$\underline{u}_d(t_i) = \begin{bmatrix} \dot{\alpha}(t_i + \frac{\Delta t}{2}) \\ \dot{\beta}(t_i + \frac{\Delta t}{2}) \end{bmatrix} \quad (56)$$

where

$$\Delta t = t_{i+1} - t_i \quad (57)$$

Equation (56) is evaluated at the interval midpoint as an approximation to

$$\int_{t_i}^{t_{i+1}} \dot{a}(\tau) d\tau \cong \dot{a}(t_i + \frac{\Delta t}{2}) \Delta t \quad (58)$$

$$\int_{t_i}^{t_{i+1}} \dot{b}(\tau) d\tau \cong \dot{b}(t_i + \frac{\Delta t}{2}) \Delta t \quad (59)$$

The matrices of equation (54) are

$$\underline{\Phi}(t_{i+1}, t_i) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{-a\Delta t} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{-b\Delta t} & \Delta t e^{-b\Delta t} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{-b\Delta t} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-a\Delta t} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-b\Delta t} & \Delta t e^{-b\Delta t} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-b\Delta t} \end{bmatrix} \quad (60)$$

The 3 x 3 submatrices in $\underline{\Phi}$ are identical and propagate the atmospheric noise states for both azimuth and elevation. See Appendix A and Ref 5 for further details of these submatrices. \underline{B}_d of (54) is specified as

$$\underline{B}_d = \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (61)$$

See Appendix D for details of the \underline{Q}_{Ad} matrix.

Now, all that remains is to define $\dot{\alpha}(t)$ and $\dot{\beta}(t)$. Missile position and velocity equations were programmed in an inertial xyz coordinate system. This was done to ease the transition from a desired trajectory to simulation equations representing that trajectory. The truth model then takes these inertial quantities and computes azimuth velocity and elevation velocity.

Figure 8 shows the geometry involved in the azimuth velocity equation. This view is looking down on the earth surface.

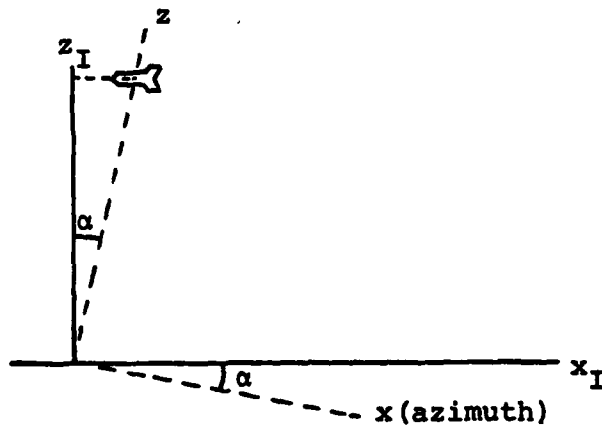


Figure 8. Azimuth Geometry

Since

$$\alpha(t) = \tan^{-1} \frac{x_I(t)}{z_I(t)} \quad (62)$$

then

$$\dot{\alpha}(t) = \frac{z_I(t)\dot{x}_I(t) - x_I(t)\dot{z}_I(t)}{z_I(t)^2 + x_I(t)^2} \quad (63)$$

yielding an $\dot{\alpha}(t)$ in rads/sec. The truth model uses a pixel as the unit of angular displacement rather than radians (1 pixel = 20 μ rads). Therefore, $\dot{\alpha}(t)$ is converted to pixels/sec by dividing by 20×10^{-6} rads/pixel.

Elevation velocity is computed similarly. Figure 9 shows the geometry involved.

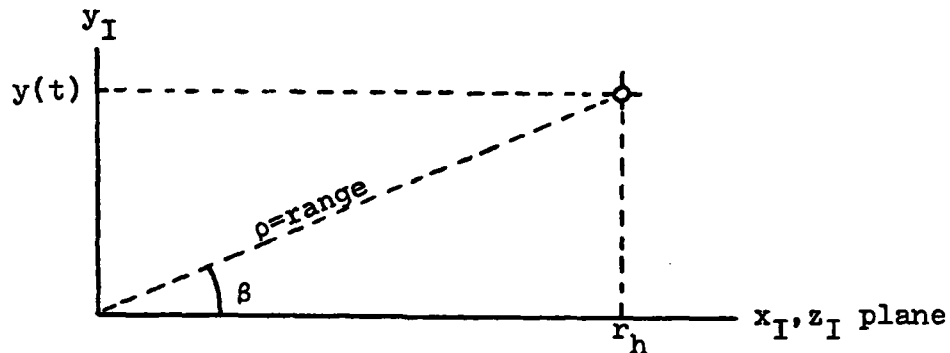


Figure 9. Elevation Geometry

where $\rho = \text{range} = [x_I(t)^2 + y_I(t)^2 + z_I(t)^2]^{1/2}$

$r_h = \text{horizontal range in } x_I, z_I \text{ plane}$
 $= [x_I(t)^2 + z_I(t)^2]^{1/2}$

Elevation can be computed using either a sine or tangent function. The tangent function was chosen as it was a simpler equation to implement using quantities already evaluated.

Since

$$\beta(t) = \tan^{-1} \frac{y_I(t)}{r_h(t)} \quad (64)$$

then

$$\dot{\beta}(t) = \frac{r_h(t)\dot{y}_I(t) - y_I(t) \left\{ \frac{x_I(t)\dot{x}_I(t) + z_I(t)\dot{z}_I(t)}{r_h(t)} \right\}}{\rho(t)^2} \quad (65)$$

Again this is rads/sec and is converted to pixels/sec as before in the azimuth velocity equation.

With $\dot{\alpha}(t_i + \frac{\Delta t}{2})$ and $\dot{\beta}(t_i + \frac{\Delta t}{2})$ inserted into equation (56) the truth model propagation equation is complete.

The trajectories simulated by the above equations were chosen to exercise and verify several aspects of filter performance. The profile used to evaluate the filter of Ref 5 is shown in Figure 10. The view is looking down toward the earth surface.

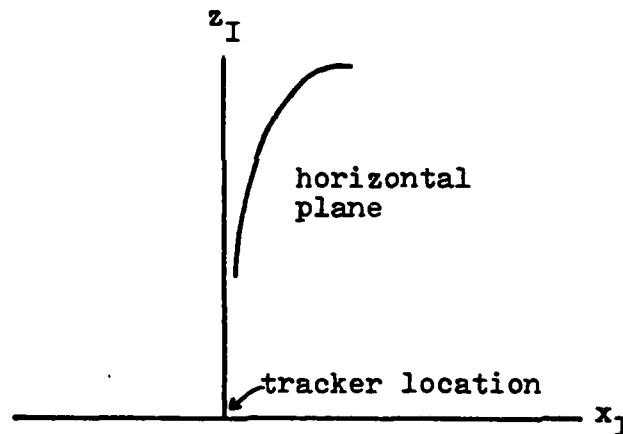


Figure 10. Trajectory 1: Approaching Target

The velocities were kept small so as not to leave the FOV in one sample period. The equations are

$$\dot{\alpha}(t_i) = \dot{\alpha}_0 e^{-a(t_i)} \quad (66)$$

$$\dot{\beta}(t_i) = 0 \quad (67)$$

where a is a suitable time constant. This was a simple trajectory that the filter of Ref 5, with no prediction capability, could track.

The first trajectory used to test the filter developed in this research was a constant velocity cross range path. Figure 11 shows this path.

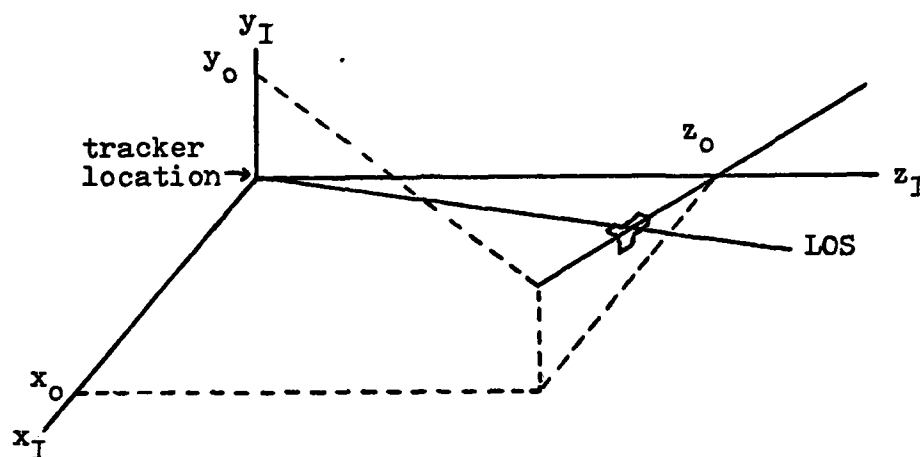


Figure 11. Trajectory 2: Cross Range Target

The equations are

$$\dot{x}_I = -500 \text{ m/sec} \quad (68)$$

$$\dot{y}_I = -300 \text{ m/sec} \quad (69)$$

$$\dot{z}_I = 0 \quad (70)$$

$$x_0 = 1500 \text{ m} \quad (71)$$

$$y_0 = 500 \text{ m} \quad (72)$$

$$z_0 = 10,000 \text{ m} \quad (73)$$

This path is relatively benign and was designed to check basic filter performance.

The next trajectory was a constant velocity cross range path with a 20g pull up at closest approach to tracker position. Figure 12 shows this path.

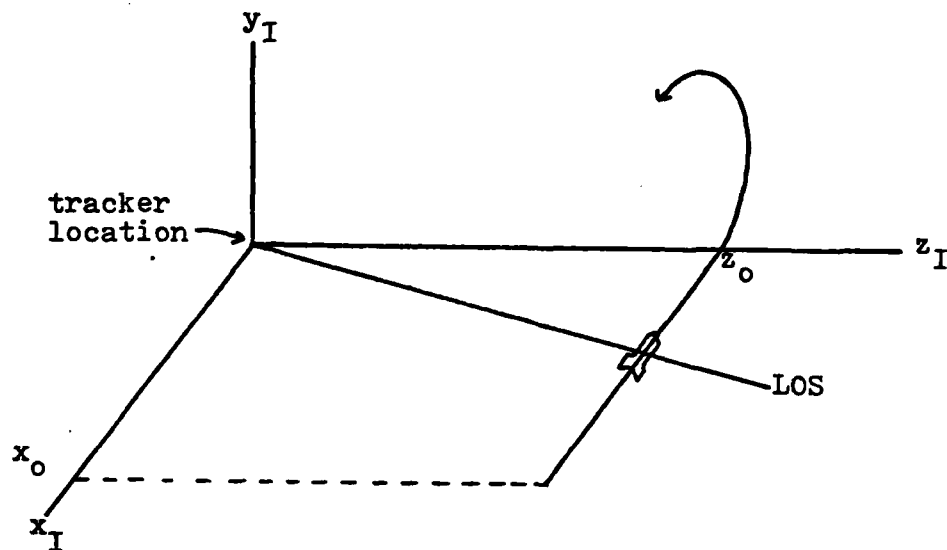


Figure 12. Trajectory 3: 20g Turn

Before the pull up the equations are

$$\dot{x}_I = -1000 \text{ m/sec} \quad (74)$$

$$\dot{y}_I = 0 \quad (75)$$

$$\dot{z}_I = 0 \quad (76)$$

$$x_O = 4000 \text{ m} \quad (77)$$

$$y_O = 0 \quad (78)$$

$$z_O = 40,000 \text{ m} \quad (79)$$

After pull up initiation the velocity equations are

$$\dot{x}_I = -1000 \cos[1.96(t-4)] \quad (80)$$

$$\dot{y}_I = 1000 \sin[1.96(t-4)] \quad (81)$$

$$\dot{z}_I = 0 \quad (82)$$

This path was intended to show the ability to track a highly maneuvering missile.

Temporally Correlated Background Noise

The analysis of real data for temporal noise correlations, which is described in detail in Appendix C, was inconclusive except for establishing the existence of such correlations. The primary problem in arriving at conclusive results was the lack of real data to analyze with backgrounds other than clear sky. The results of the work accomplished (Table C-I) demonstrate a consistent temporal correlation coefficient from one sample time to the next of about .364 for the clear sky. This value for a time difference of one thirtieth of a second, while disappointingly low, was not unexpected considering the back-ground, which had essentially no infrared structure to be correlated in time.

In the absence of guidance from the data analysis, the formulation of a model for the temporal correlation was based on past models and ease of implementation. Depending upon the model chosen, the representation of the temporal correlations required one or more states for each measurement pixel, a minimum of 64 new states to add to the truth model. A first order model was chosen because analysis had not indicated the need for a higher order model and the number of states would be minimized. The exponential time correlation form was assumed to be an appropriate model which is easily handled conceptually and often used in temporal modelling.

The stationary exponentially correlated model of the relationship between the background noises of a particular pixel at times t and s incorporates the following statistics:

$$E\{v(t)\} = E\{v(s)\} = 0 \quad (83)$$

$$E\{v(t)v(s)\} = \sigma_N^2 e^{-|t-s|/\tau_N} \quad (84)$$

where σ_N = the desired variance of the noise

τ_N = the correlation time constant of the noise

The continuous time equation for generating $v(t)$ for a single pixel is represented in Figure 13 and written as

$$\dot{v}(t) = -1/\tau_N v(t) + w_N(t) \quad (85)$$

where w_N is zero mean white Gaussian noise whose variance kernel must be (Ref 8:178)

$$E\{w_N(t)w_N(t+\tau)\} = \frac{2\sigma_N^2}{\tau_N} \delta(\tau) \quad (86)$$

so that the variance of $v(t)$ is σ_N^2 .

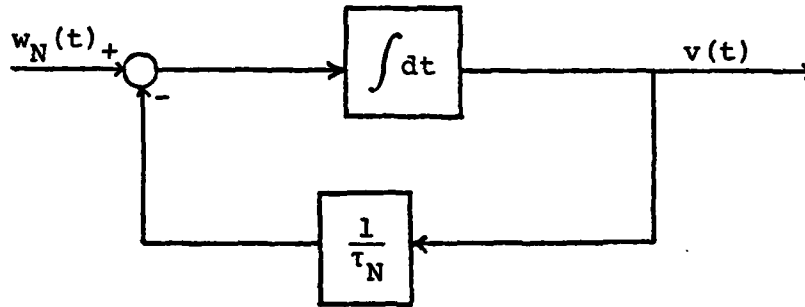


Figure 13. Continuous First Order Noise Generation Model

The equivalent discrete time form of this model (continuous to discrete time conversions is covered in Ref 8, pages 170 to 174) is shown in Figure 14 and written as

$$v(t_{i+1}) = \phi_N(t_{i+1}, t_i) v(t_i) + w_{Nd}(t_i) \quad (87)$$

where

$$\phi_N(t_{i+1}, t_i) = \mathcal{L}^{-1}\{[s + \frac{1}{\tau_N}]^{-1}\} = e^{-(t_{i+1}-t_i)/\tau_N} \quad (88)$$

and

$$E\{w_{Nd}(t_i)\} = 0 \quad (89)$$

$$E\{w_{Nd}^2(t_i)\} = \sigma_N^2(1 - e^{-2(t_{i+1}-t_i)/\tau_N}) \quad (90)$$

in order to generate $v(t_i)$ with appropriate statistics. For computer simulation, the process to generate $v(t_{i+1})$ is shown in Figure 15. $w(t_i)$ is a zero mean, unit variance, white discrete-time Gaussian noise generated by a Gaussian noise generator.

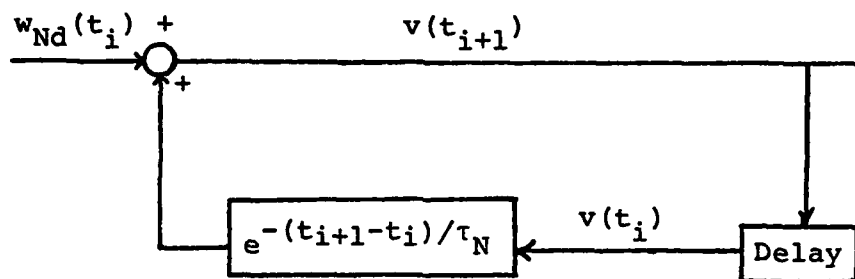


Figure 14. Discrete First Order Noise Generation Model

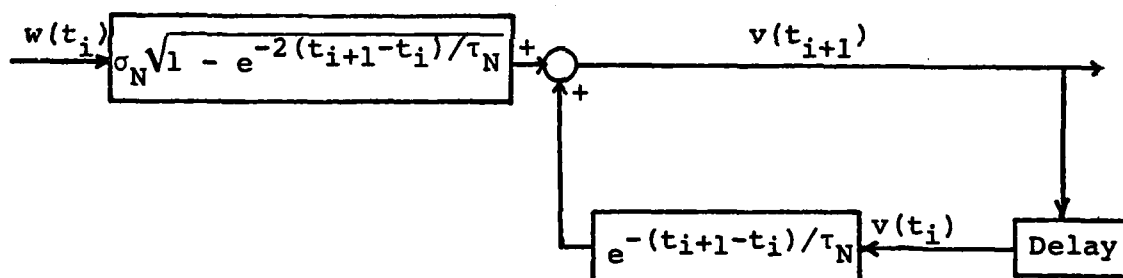


Figure 15. Discrete First Order Noise Generation Model Driven by Unit Variance Noise

Since the 64 noise states are independent of each other, the equations for $\underline{v}(t_{i+1})$, a vector of dimension 64, is easily written as

$$\underline{v}(t_{i+1}) = \underline{\phi}_N(t_{i+1}, t_i) \underline{v}(t_i) + \underline{w}_{Nd}(t_i) \quad (91)$$

where $\underline{\phi}_N(t_{i+1}, t_i)$ is a 64-by-64 diagonal matrix written as:

$$\underline{\phi}_N(t_{i+1}, t_i) = e^{-(t_{i+1}-t_i)/\tau_N} \underline{I} \quad (92)$$

The covariance of $\underline{w}_{Nd}(t_i)$ is diagonal matrix of dimensions 64 by 64

$$E\{\underline{w}_{Nd}(t_i)\underline{w}_{Nd}^T(t_{i+1})\} = \underline{R} = \sigma_N^2(1 - e^{-2(t_{i+1}-t_i)/\tau_N})\underline{I} \quad (93)$$

Since these states are decoupled from the other truth model states described in (54), the 64 noise states can be augmented as a block onto the end of the original eight state truth model, yielding a 72 state truth model.

The generation of background noise which is spatially correlated as well as temporally correlated is accomplished by allowing the driving noise of the 64 shaping filters to be correlated. This is simply accomplished by inserting the appropriate nonzero values in the off-diagonal terms in (93) above. The modelling and computation of \underline{R} for spatially correlated noises is detailed in a preceding section titled Spatially Correlated Background Noise. This formulation assumes the separability of the spatial and temporal effects in the background noise.

The implementation of the temporally correlated background noises requires appropriate state rearrangement (index changes) to compensate for the apparent motion of the background due to target motion. When tracking a target moving horizontally from left to right, the background will appear in the FLIR to move with equal velocity in the opposite horizontal direction (right to left). To keep the simulated background of the target correct, three things must be done: the target motion for each sample period must be computed, the temporal noise states must be moved to reflect the target motion, and those states representing pixels with new background must be filled with new spatially correlated noises (not time correlated with previous state values because the previous history of these pixels is not known). These three steps make up three subroutines in the simulation software which can be seen in Appendix F. The pixel

motion taken precisely from the other truth model states is rounded to the nearest integer pixel before the state rearrangements are performed. The motion was restricted to integer pixel values to allow the same 8-by-8 array model to be used at all times easily. The rounded motion value is a fraction of a pixel in error from the true motion value. This fraction is stored for one sample period and added to the next motion value before the rounding process is executed. This avoids an accumulation of error due to the rounding process. This approximation was deemed valid for target velocities of significant size, thirty or more pixels per second (or for motionless targets, 0 pixels per second, when no state transitions are needed).

In summary, a step by step algorithm for the generation of temporally and spatially correlated noises is as follows.

- (1) \underline{w} , a 64-dimensional vector of zero mean, unit variance, white Gaussian noises is generated.
- (2) \underline{w}_{Nd} is computed by premultiplying \underline{w} with $\sqrt{\underline{R}}$, the Cholesky square root of \underline{R} .
- (3) The states \underline{v} are propagated forward in time by multiplying by $\underline{\Phi}_N(t_{i+1}, t_i)$ and then adding \underline{w}_{Nd} .
- (4) The pixel motion for the time step is computed to the nearest pixel and the ignored fractions are stored.
- (5) The states are moved appropriately.
- (6) Those states now representing new background are filled with new noises.

Field Size

The FLIR field of view (FOV) is a fixed angle and in this research was taken to be 160 μ rads wide. Although arbitrarily chosen, this is close to the values AFWL is using. As a vehicle moves toward or away from the FLIR, the

angle subtended by the image increases or decreases. At some range the image will fill the FOV. For example, at 10 km a vehicle 1.6 m long will fill the FOV. Clearly, this is inadequate for tracking air-to-air missiles. Two alternatives seem readily apparent, a larger FOV or compressing the image to stay within the existing FOV. That is, combine a 2 x 2 array (4 elements) into one pixel, etc.

Larger FOV seems attractive except that it would increase the computational burden significantly. Increasing the FOV from the current 8 x 8 pixel array to, say, 16 x 16 will quadruple the computations required in some parts of the filter algorithm. When computation time becomes a large part of the sample period, the assumption that control outputs occur almost instantaneously with sample time is no longer valid. This may introduce instability into a system which otherwise would be stable.

Compressing the image will keep it within the FOV but at the expense of resolution. However, if the tracking system works well, the loss in accuracy may be tolerable. Two compression simulation approaches were considered: (1) have the truth model actually compute a 16 x 16 array to present to the filter for 2:1 compression, or (2) reduce the truth model sigma's by some compression factor but still present the filter an 8 x 8 array. Either way would have required extensive programming while not furthering the main thrust of this research. Therefore, image dimensions were maintained within the FOV to avoid the need to confront the compression issue.

Although not done here, consideration of FOV compression or expansion is an area which is a logical extension of this research.

Referencing

REFERENCING, which is a tracking simulation package supplied by AFWL, was used primarily for analyzing the

temporal correlations of the background noise in real data as described previously. However, REFERENCING has considerable potential to be used as an alternate truth model and as a more extensive analysis tool.

REFERENCING has two operating modes: one mode uses real digital target data from magnetic tape and the other mode generates its own synthetic target data. In either mode the real target or synthetic target is tracked by one of six correlation tracking schemes currently being studied by AFWL (Ref 13:1-2). Starting with a 16-by-16 reference array containing an edge of the target from an initial data frame, REFERENCING computes a 16-by-16 (or smaller) matrix of cross-correlations between the reference array and sub-matrices of a search matrix in a subsequent data frame. (The cross correlation array was used in temporal noise analysis in Appendix C.) The cross correlation matrix is then utilized by the correlation trackers to compute a precise target offset between the two frames.

Compared to the truth model generated in this study, REFERENCING has one serious drawback as a good representation of the real world. In terms of target size and shape, and the noise characteristics of the measurements, the real digital data from tape cannot be surpassed by a software model of the target and noises. However, because the real data was taken from the FLIR which was moving to keep the target centered in its field of view, the target appears essentially motionless. Moreover, the true motion (states) of the target is not discernible from the data tapes. Thus, the errors committed by any tracking system could not be computed.

When operated in the synthetic target mode, REFERENCING is deficient in representing any real target motion. The synthetic target can be generated in almost any size and shape with its edges smoothed (Gaussian shape) by a filter; in terms of target intensity characteristics it is a good

simulation. The target can then be either segmented (rows or columns of the target deleted), rotated, or jittered during each run. However, the target cannot display real motion. Thus, the target motion is again lacking in realism, and except when the target is jittered there is no noise in the synthetic target frames.

With help from AFWL, the REFERENCING routine could be made into an excellent "truth model" using real digital data. The target motion could be recreated from the time history of the FLIR pointing angles and any other target data available (i.e. radar range and velocity) from AFWL. The true trajectory could not be computed, but an adequate noise-corrupted representation probably could be achieved.

III. Previous Filter Robustness

Introduction

The robustness of the original filter is investigated when it is subjected to truth model parameter mismatches and to major truth model changes. The first group of computer simulations, using the original truth model and filter, investigated the filter performance when parameter mismatches existed between the truth model and the filter. A second group of simulations evaluated the performance of the original filter when subjected to the new target scenario and background as depicted by major truth model changes described in Chapter II.

Before discussing the particular computer simulation runs, the general simulation scenario will be briefly outlined. Monte Carlo simulations were performed on a CDC 6600 using software packages written in FORTRAN IV. Each Monte Carlo case consisted of 20 five second simulations. Each of the 20 simulations was started at the same specific value of $\underline{x}_0(0)$ and $P_0(.5I)$. The data from the Monte Carlo runs was later processed and appropriate statistics for the atmospheric jitter and target state estimate errors were plotted on graphs. Appendix E discusses in more detail notable computer simulation considerations such as noise generation, approximations, and tuning. Appendix F contains copies of the FORTRAN IV software packages used to perform the Monte Carlo simulation. Appendix G contains a copy of the software module which computed the simulation statistics and plotted the data. Appendix H lists the parameters for cases 1-31 which were performed as part of the analysis for this chapter. Appendices I and J contain the plots which display the detailed results of the simulations contained in this chapter; major results, trends, and discussions will be presented in the chapter itself.

Sensitivity to Parameter Mismatches

The performance of the previously designed filter was evaluated by inputting contrasting parameters for the filter and truth models. Two preliminary baseline cases were run. Case 1 was run at a signal to noise ratio of 20 ($S/N = I_{\max}/\sigma_N$) and served as a comparison to previously reported results (Ref 5). Case 2 was run as a baseline for cases where $S/N=10$. With a nominal set of parameters, Case 2 serves a performance standard for the other runs. Cases 3 through 12 duplicate case 2 except one parameter in the truth model was changed in each case. Five areas of sensitivity were examined by giving the truth model high and low values of a parameter while the filter was given the nominal value of the parameter. The areas of interest were target dynamics correlation time, target intensity, target size, the root mean square (rms) value of background noise, and the ratio of the target dynamics rms value to the atmospheric jitter rms value.

Case 1, whose input parameters can be found in Appendix H and resulting plots can be found in Appendix I, was run to check out the inherited software packages through comparison to Case 1 conducted by Captain Mercier (Ref 5:47). Both cases were run at a signal to noise (S/N) ratio of 20, and they agreed very closely in final results (see Appendix I and Ref 5:96 to 98). Mercier's Case 1 was not duplicated exactly due probably to a difference in the seeds for the random number generator. However, the variances of the real error and of the filter estimated error agreed very closely.

Case 2, which was conducted at a signal to noise ratio of 10, serves as a baseline for comparison for the rest of the mismatch cases. Case 2 was executed with all the nominal parameters, no mismatched parameters. Table I summarizes the results of case 2 - the units are pixels for the average mean absolute error and for the average standard deviation. Because all the errors tend to be zero mean,

TABLE I
Baseline for Cases 3-18: Case 2 Results

	x	y
Average Standard Deviation		
Atmospheric Jitter		
Actual	.55	.58
Filter-Indicated	.58	.58
Target Position		
Actual	.56	.58
Filter-Indicated	.58	.58
Average Absolute Error		
Atmospheric Jitter	.07	.08
Target Position	.10	.12

the average mean absolute error proved to be more meaningful and is presented in the tables to follow. Each value shown represents the average of the value over the five second simulation period. Figures 16, 17, and 18 are three of the plots resulting from case 2. The three plots display statistics for the target dynamics in the x direction. Similar plots were made for the x-direction atmospheric jitter, the y-direction target dynamics, and the y-direction atmospheric jitter; they can be seen in Appendix I. Figure 16 demonstrates the true error standard deviation (σ) convergence of the target dynamics estimate. The true error standard deviation (σ) is plotted as a function of the number of runs to demonstrate the convergence which occurs when 20 simulations are run. Figure 17 is a very effective tool for tuning analysis as it displays the true error standard deviation (the noisy curve) as well as the filter-indicated error standard deviation (the more consistent curve), good tuning being achieved typically when these agree. Figure 18 displays the mean true target position error for the 20 Monte Carlo runs and the one sigma envelope of the errors for performance analysis.

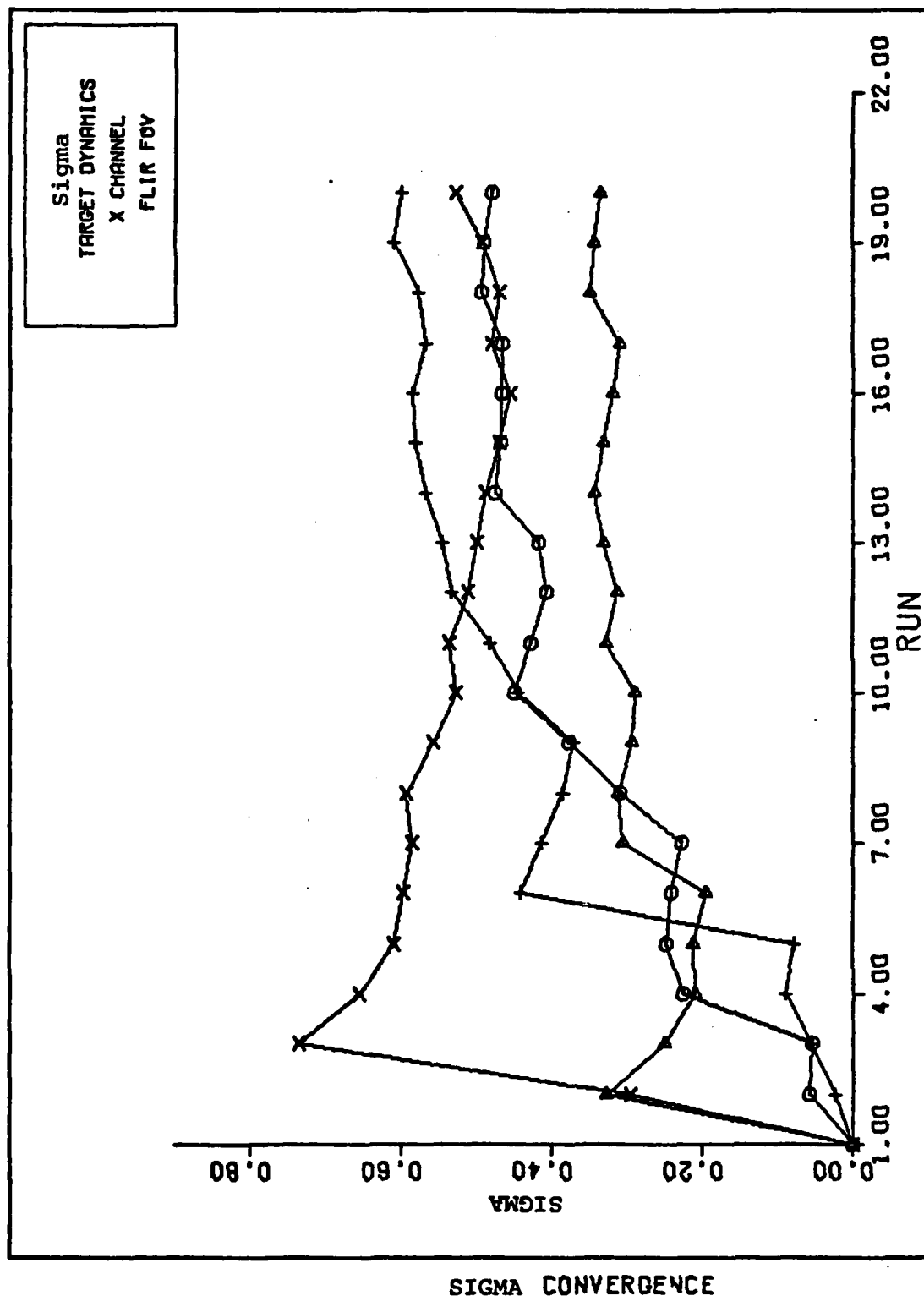
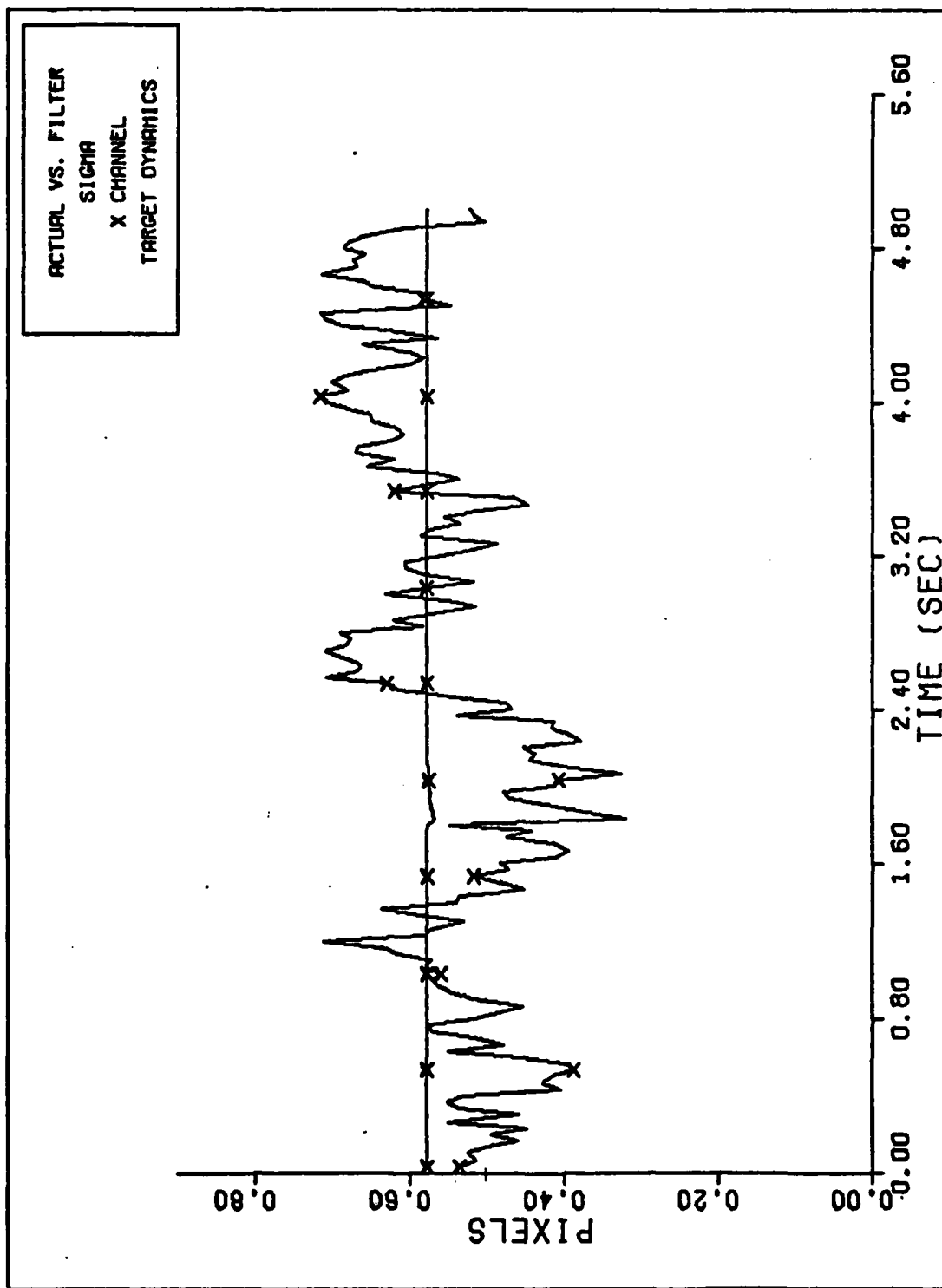
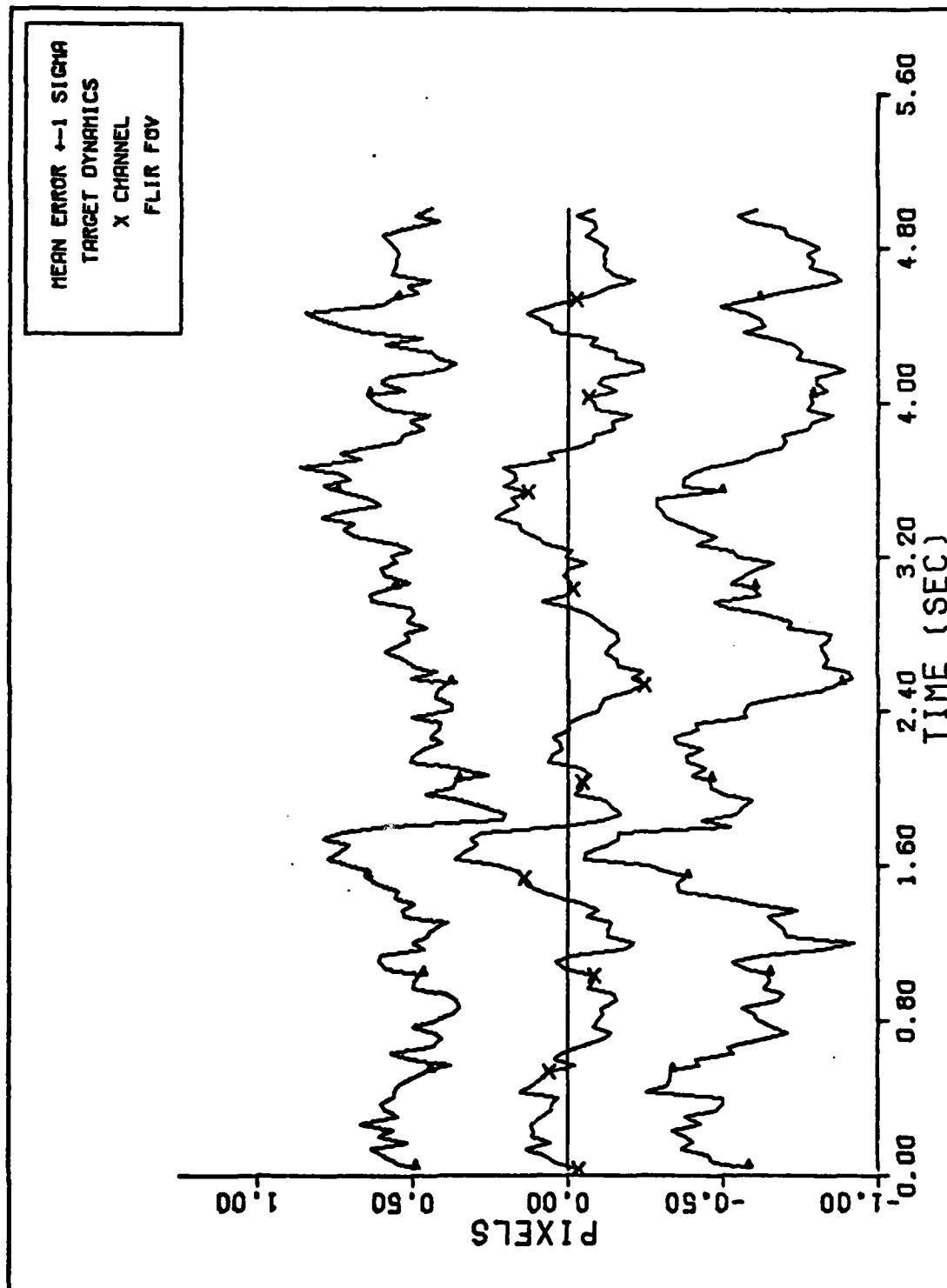


Figure 16. Case 2: Target Position x Channel Standard Deviation (σ , Sigma) Convergence



FILTER VS. ACTUAL SIGMA PLOT (S/N = 10)

Figure 17. Case 2: Target Position, X Channel, Real Error versus Filter Standard Deviation



X CHANNEL DYNAMICS ERROR (S/N= 10)

Figure 18. Case 2: Target Position, X Channel, Mean Error and 1 Sigma Envelope

The first area of interest for studying filter robustness was variations in target dynamics. The nominal value for τ_D in equations (3) and (4) was set at 1 sec. For case 3 the filter assumed $\tau_D = 1$, but the truth model was based on $\tau_D = 0.2$. This can be interpreted as the target exhibiting higher frequency motion (more jittery) than the filter assumes. For case 4, the truth model is based on $\tau_D = 5$, a slower, more time-correlated motion than the filter expects. The results for both cases are shown in Table II and a limited set of their plots can be seen in Appendix I. The filter performance for both cases is very acceptable. However, the results have little meaning in terms of the air-to-air missile scenario because the truth model's target motion equation for these cases is extremely benign.

Cases 5 and 6 test the filter's response to mismatches in maximum target intensity. The nominal maximum target intensity (I_{\max}) measure of 10 (intensity units, the exact units are irrelevant to the purpose of this as long as appropriate ratios are maintained) was used by the filter with the rms value of the background noise set at 1 (intensity unit). For case 5 the truth model is given $I_{\max} = 1$; the real FLIR images show a target image that is very much masked by the noise while the filter thinks that the target image should be readily distinguishable. The truth model for case 6 has $I_{\max} = 20$. Now the target intensity is greater than the filter model assumes. The performance of the filter in both cases is not acceptable as can be seen in Table III and their plots in Appendix I. While the poor performance of case 5 is hardly surprising (such a low S/N can cause poor performance with or without a mismatch), the fact that case 6 exhibited similar results is very interesting. The filter appears to have difficulty because it is searching for the wrong shape of intensity profile caused by the mismodelled I_{\max} . These results should generally be applicable to the close range air-to-air missile problem because both scenarios share the Gaussian intensity model.

TABLE II

Target Dynamics Mismatch -
Case 3: Low τ_{DT} and Case 4: High τ_{DT} Results

	Case 3 $\tau_{DT} = .2$		Case 4 $\tau_{DT} = 5$	
	x	y	x	y
Average Standard Deviation				
Atmospheric Jitter				
Actual	.65	.65	.48	.45
Filter-Indicated	.58	.58	.58	.58
Target Position				
Actual	.65	.68	.45	.42
Filter-Indicated	.58	.58	.58	.58
Average Mean Absolute Error				
Atmospheric Jitter	.10	.12	.10	.07
Target Position	.11	.11	.11	.09

The third set of mismatches that were studied dealt with target size. The nominal, filter, value for the target size was three pixels ($\sigma_g = 3$). In case 7 the truth model had the target size set at 1 pixel ($\sigma_g = 1$); the real target is much smaller than the target the filter is trying to identify. Consequently, the real image randomly walks in the large target envelope that the filter is searching for, with the result of poor performance, as can be seen in Table IV and the plots in Appendix I. The filter performs acceptably in case 8, in which $\sigma_g = 5$ for the truth model. The filter appears to pin down the center of the target much better if the real image is larger than the assumed filter image size, as can be seen in Table IV. These results are very meaningful for the close range target where the real image size may be not only mismatched but also changing in size and shape.

Mismatches in the background noise were studied in cases 9 and 10. The nominal value for the signal to noise

TABLE III

Target Intensity Mismatch -
Case 5: Low I_{\max} and Case 6: High I_{\max} Results

	Case 5 $I_{\max} = 1$		Case 6 $I_{\max} = 20$	
	x	y	x	y
Average Standard Deviation				
Atmospheric Jitter				
Actual	1.7	1.60	1.5	1.5
Filter-Indicated	.60	.60	.58	.58
Target Position				
Actual	3.5	4.0	.70	.70
Filter-Indicated	.65	.65	.58	.58
Average Absolute Error				
Atmospheric Jitter	.20	.40	.25	.25
Target Position	.20	.10	.10	.20

ratio ($S/N = I_{\max}/\sigma_N$) was 10, as modelled in the filter. For case 9 the truth model S/N was 1 (the variance of the background noise is 10 instead of 1 and I_{\max} equals 10 for both filter and truth model). This case is very similar to case 5; the filter is searching for a prominent target against low background noise but the real target is actually surrounded by background noise of equal intensity. In case 10, the truth model has a $S/N = 20$ so that the variance of the background noise is 0.5. Case 9 resulted in terrible filter performance (Table V) for reasons similar to those of the poor performance exhibited in case 5. For both cases, the real measurements in both portray a target image effectively hidden among the noise. This is probably the major cause of the poor performance in both cases rather than the mismatches which existed. A similar analogy does not exist between cases 6 and 10. Case 10 exhibited very good filter performance which is understandable when interpreted physically. Both the truth model and filter in case 10 have $I_{\max} = 10$. The mismatch is that the $S/N=10$ for the

TABLE IV

Target Size Mismatches -
Case 7: $\sigma_{gT} = 1$ and Case 8: $\sigma_{gT} = 5$ Results

	Case 7 $\sigma_{gT} = 1$		Case 8 $\sigma_{gT} = 5$	
	x	y	x	y
Average Standard Deviation				
Atmospheric Jitter				
Actual	1.5	1.5	.63	.62
Filter-Indicated	.58	.58	.58	.58
Target Position				
Actual	3.0	3.0	.65	.65
Filter-Indicated	.61	.62	.58	.58
Average Absolute Error				
Atmospheric Jitter	.25	.4	.10	.12
Target Position	.5	1.5	.15	.10

filter results in the variance of the noise being 1 whereas the variance of the noise in the truth model is 0.5 ($S/N=20$). Simply, the assumed target intensity shape is correct (instead of erroneous as in case 6) while the background noise is actually less than the filter assumes, and it tracks the target very well.

The final mismatch runs investigated the ratio of rms value of the target motion to rms value of the atmospheric jitter. These values are assumed to be equal for the nominal case ($\sigma_D/\sigma_A = 1$). In case 11, the rms value of the target motion was set to a fifth of the rms value of atmospheric jitter ($\sigma_D/\sigma_A = .2$). Physically interpreted, the real target motion is much less in magnitude than the filter is searching for. So the filter over estimates the errors, as can be seen in Table VI. In case 14 the real target has considerably more motion ($\sigma_D/\sigma_A = 5$) than the filter is programmed to expect, resulting in unacceptable filter performance shown again in Table VI and Appendix I. The results

TABLE V

Background Noise Mismatch -
Case 9: S/N = 1 and Case 10: S/N = 20 Results

	Case 9 S/N = 1		Case 10 S/N = 20	
	x	y	x	y
Average Standard Deviation				
Atmospheric Jitter				
Actual	8.0	8.1	.57	.56
Filter-Indicated	.63	.63	.58	.58
Target Position				
Actual	3.8	3.9	.56	.53
Filter-Indicated	.77	.78	.58	.58
Average Absolute Error				
Atmospheric Jitter	1.5	1.5	.12	.10
Target Position	.5	.6	.12	.12

of these cases again have little meaning in terms of the air-to-air missile problem. The truth model's target motion and the filter's target motion model in these cases are totally unrealistic in terms of representing an air-to-air missile. The truth model's target motion has already been redefined in Chapter II. An appropriate change in the filter's target dynamics model is therefore motivated.

Sensitivity to New Truth Model

The previously formulated filter was also evaluated as it was subjected to the major truth model changes described in Chapter II. First, the filter was tested against targets with differing aspect ratios ($AR = \text{image length/image width} = \sigma_v/\sigma_{pv}$ as defined in Chapter II) as simulated by the truth model. Next, filter performance was evaluated when the background noise was spatially correlated. The truth model was then changed to allow target motion, and the filter performance tested in terms of the new motion. Temporally

TABLE VI

Ratio of Atmospheric Jitter to Target Dynamics Mismatch -
Case 11: $\sigma_D/\sigma_A = .2$ and Case 12: $\sigma_D/\sigma_A = 5$ Results

	Case 11		Case 12	
	x	y	x	y
Average Standard Deviation				
Atmospheric Jitter				
Actual	.43	.41	1.7	1.5
Filter-Indicated	.58	.58	.58	.58
Target Position				
Actual	.38	.37	2.5	2.8
Filter-Indicated	.58	.58	.59	.60
Average Absolute Error				
Atmospheric Jitter	.08	.08	.35	.4
Target Position	.10	.07	.4	.5

correlated background noise was the last addition to the truth model, and, subsequently, the filter performance was evaluated for various target scenarios with a variety of temporal and spatial background noises. Each truth model change as formulated in Chapter II was aimed at a more realistic representation of the problem. The filter performance in light of these changes to the truth model was not expected to be good because the filter had not been designed to track the type of target which the changes were attempting to create. The goal of these runs was to obtain a prioritized list of those characteristics of the new target scenario which caused the most problems.

The filter's ability to track a target with an aspect ratio (AR) greater than one was first tested. This was accomplished by a software change to replace the circular constant intensity contour target by a target (also bivariate Gaussian) with variable minor (σ_{pv}) and major ($\sigma_{pv} \cdot AR$) axes' magnitudes; target dynamics and atmospheric jitter state equations in the truth model remained as originally

formulated. For the cases tested, the major axis was aligned with the x-axis. Cases 13, 14, and 15 were run with σ_{pv} equal to 1 and AR equal to 2, 5, and 10, respectively. The results of these cases are shown in Table VII. (The plots and input parameters for the truth model and filter for each case are found in Appendices H and I, respectively.) As expected in all cases, the y-axis filter performance was very good because the target size did not change in that direction. In the x-axis direction, the filter performance did not degrade significantly as the target increased in aspect ratio. These results show small but definite errors which would be amplified by other complicating factors such as feedback, target rotation, target motion, and variations in target size. Other factors and simulations need to be considered before judging the role of AR.

The truth model changes, discussed in Chapter II, for creating spatially correlated background noise were implemented, and the filter was tested against it. Again, the state equations were not changed, and the circular constant intensity contour target was used (AR = 1). First and second spatial correlations using the exponential model were used. The 24 correlation coefficients of any one pixel have only five different values due to the symmetry of the correlations. For case 16 for example, $r_{k,k+1}$, $r_{k,k+9}$, $r_{k,k+2}$, $r_{k,k+10}$, and $r_{k,k+18}$ shown in Figure 5 were set to .368, .243, .135, .105, and .059, respectively, as generated by an exponential correlation model. The other 19 coefficients were set equal to one of these values as dictated by the symmetry. Cases 17 and 18 were run with spatial correlations of .497, .372, .247, .209, and .138 at S/N and I_{\max} equal to 10 and 1, respectively. These coefficients represent the upper limit of values which could be used without numerical problems in R (see Chapter II).

TABLE VII

Elliptical Target Intensity Contours in the Truth Model
Cases 13, 14, and 15 Results

	Case 13 AR = 2		Case 14 AR = 5		Case 15 AR = 10	
	x	y	x	y	x	y
Average Standard Deviation						
Atmospheric Jitter						
Actual	.65	.58	.82	.58	.85	.56
Filter-Indicated	.58	.58	.58	.58	.58	.58
Target Position						
Actual	.60	.55	.80	.58	.90	.54
Filter-Indicated	.58	.58	.58	.58	.58	.58
Average Absolute Error						
Atmospheric Jitter	.15	.10	.20	.12	.20	.12
Target Position	.10	.10	.15	.10	.15	.10

Note: the baseline, Case 2, for comparison
is summarized in Table I.

Cases 16, 17, and 18 can be interpreted physically as tracking a nearly stationary target against a spatially correlated background while the filter assumes that the background noise is spatially white. The results of these three cases, which are displayed in Table VIII, showed acceptable performance except for case 18 when $S/N = 1$. The filter performance in cases 16 and 17 was almost identical to case 2; the spatially correlated background noise caused no perceived degradation. Case 18's poor performance was probably caused equally by the low signal to noise ratio and the spatial correlations in the background noise. This was partially confirmed by comparison to case 10 of Capt. Mercier's thesis (Ref 5:124-125) which shows a true error variance for the atmospheric jitter and target position of 1.0 and 1.25, respectively (even here the filter underestimated its own errors). This case corresponds to case 18

TABLE VIII

Spatially Correlated Background Noise in the Truth Model
Cases 16, 17, and 18 Results

	Case 16		Case 17		Case 18	
	S/N = 10		S/N = 10		S/N = 1	
	Low Cor-relations		High Cor-relations		High Cor-relations	
	x	y	x	y	x	y
Average Standard Deviation						
Atmospheric Jitter						
Actual	.58	.57	.58	.58	1.5	1.55
Filter-Indicated	.58	.58	.58	.58	.85	.85
Target Position						
Actual	.55	.56	.58	.56	2.5	2.2
Filter-Indicated	.58	.58	.58	.58	.65	.65
Average Absolute Error						
Atmospheric Jitter	.13	.12	.12	.10	.30	.30
Target Position	.08	.10	.08	.11	.40	.50

Note: the baseline, Case 2, for comparison
is summarized in Table I.

here, but with no spatial and temporal correlation in the truth model background noise, i.e. with no filter mismatch.

The next test for the filter was the addition of motion to the truth model (refer to Chapter II for description of target motion formulation). The elliptical target and spatial background noise were not used for these test cases. One change was made to the filter to allow it to try to track a moving target. The original filter was open loop in design; it was assumed that the computed offsets were not zeroed out by the system's control mechanism. Thus, the filter relied on the zero mean characteristic of the original truth model's dynamics to keep the target in the FLIR field of view. Without implementing feedback, the filter would lose any moving target in one or two sample periods

depending on the magnitude of the motion. (For more detailed analysis see Chapter IV.) Thus, the filter was changed to reflect control system zeroing of the computed target motion offsets for each sample period. The velocities for cases 21 and 22 were 56.4 pixels per second ($x = 40$ pixels/sec, $y = 40$ pixels/sec, initially) and 84.6 pixels per second ($x = 60$ pixels/sec, $y = 60$ pixels/sec, initially), respectively.

Because of the software change to provide system feedback, a new baseline case was computed. The parameters and plots for case 19 can be found in Appendices H and J, respectively. Case 19 was run with a signal to noise ratio of 10 and without target motion ($\dot{x} = 0$, $\dot{y} = 0$). The results of case 19 are summarized in Table IX. The results show the filter to be slightly out of tune (tuning is discussed more completely in Appendix E). Tuning was not accomplished because of time constraints and the small amount the filter was mistuned. This is partially because of the feedback change and partially due to a change in two of the nominal parameters ($\sigma_{DT}/\sigma_{AT} = 5$ and $\sigma_{AF} = .2$). Case 19 will be the baseline of comparison for cases 20 to 23.

Table IX and subsequent tables in this chapter, a new analysis value has been added - the number of times in the 20 simulations that the target was lost. If the true target position ever was more than three times the target size ($3 \cdot \sigma_v$) away from the filter computed position, the target was considered lost. For targets where σ_v is greater than or equal to 1.5 pixel, this means most of the target image has left the field of view. For any size target this criteria means that the true image is so far from the predicted image that the residuals provide little useful information. In most cases where the target is lost more than one or two times, the total number of loss of track is more meaningful than the statistics that result.

TABLE IX

Baseline for Cases 20-23
Case 19 Results
(S/N=10)

	Case 19	
	x	y
Average Standard Deviation		
Atmospheric Jitter		
Actual	.25	.25
Filter-Indicated	.19	.19
Target Position		
Actual	.40	.40
Filter-Indicated	.19	.19
Average Absolute Error		
Atmospheric Jitter	.09	.08
Target Position	.10	.10

Loss of track - 0 times

The filter performance for cases 20, 21, and 22, summarized in Table X, show a trend of increasingly poor performance corresponding to the increasing velocity of the cases. Case 22 statistics are not shown because the loss of track problems caused numerical problems serious enough to interrupt its execution. It is known that loss of track occurred in all 20 simulations for case 22. The results of these three cases demonstrate all too well the need for better filter modelling of the target motion.

The last truth model change was the formulation of a 72 state truth model to allow temporal noise modelling of the background noise. The filter performance was analyzed in cases of various combinations of S/N, temporal correlations, and spatial correlations. To make sense physically, temporal and spatial correlations were used together in the background noise for most cases.

The first two cases, cases 23 and 24, were run at S/N equal to 10 and 1, respectively, with both spatial and

TABLE X

Target Motion in the Truth Model
Case 20 and 21 Results

	Case 20		Case 21	
	$\dot{x} = 20$ P/S $\dot{y} = 20$ P/S		$\dot{x} = 40$ P/S $\dot{y} = 40$ P/S	
	x	y	x	y
Average Standard Deviation				
Atmospheric Jitter				
Actual	.26	.28	.22	.23
Filter-Indicated	.19	.19	.19	.19
Target Position				
Actual	.45	.50	8.0	16.0
Filter-Indicated	.20	.20	.20	.20
Average Absolute Error				
Atmospheric Jitter	.12	.14	.11	.15
Target Position	.20	.22	7.0	15.0
Loss of Track	0 times		12 times	

Note: 1) Case 22 lost track 20 times.
2) The baseline, Case 19, for comparison is summarized in Table IX.

temporal correlations. For each case the spatial correlation coefficients were the same as those used in case 16; the truth model depicted a motionless target. For the temporal noise τ_N was chosen ($\tau_N = 20$ sec) to simulate a .95 correlation between noises over one second of time. This value of τ_N is probably a realistic upper limit (worst case) on the correlations; a correlation of 1 is the theoretical upper limit. Filter performance in case 23 shown in Table XI was only slightly worse than that of the baseline, case 19, and acceptable. The performance of case 24 was so bad that numerical problems halted the simulation software. These two cases reaffirmed past conclusions: at S/N equal

TABLE XI

Temporal and Spatial Background Noise in
Truth Model (SN = 10) - Case 23 Results

	Case 23 .95 Time Cor.	
	x	y
Average Standard Deviation		
Atmospheric Jitter		
Actual	.26	.27
Filter-Indicated	.20	.20
Target Position		
Actual	.50	.49
Filter-Indicated	.19	.19
Average Absolute Error		
Atmospheric Jitter	.09	.08
Target Position	.15	.12
Loss of Track	0 times	

Note: 1) Case 24 loss of track = 20 times.
2) The baseline, Case 20, for comparison is summarized in Table IX.

to 10, no matter what realistic model used, the background noise has no significant effect on filter performance; at S/N equal to 1 real tracking problems occur.

A group of cases, simulated with S/N equal to 2, provided more interesting evaluations of the temporal and spatial background noise correlations effect. A new baseline, case 25, was computed because of the new S/N. Case 25 was run without truth model motion and with temporal and spatial correlations in the background noise deactivated (actually $\tau_N = .01$ sec which corresponds to a .03 correlation in one sample period). Case 25 results are tabulated in Table XII. While the performance statistics are not good, loss of track only occurred once, and these values will serve for comparison purposes for the cases to follow.

TABLE XII

Baseline for Cases 26-29: Case 25 Results
(S/N=2)

	Case 25	
	x	y
Average Standard Deviation		
Atmospheric Jitter		
Actual	.20	.19
Filter-Indicated	.19	.19
Target Position		
Actual	1.9	1.3
Filter-Indicated	.34	.34
Average Absolute Error		
Atmospheric Jitter	.05	.05
Target Position	.40	.30
Loss of Track	1 time	

Cases 26, 27, 28, and 29 were run at S/N equal to 2 and various combinations of temporal and spatial background noise. Case 26 had high temporal correlations (.95 for one second) and spatial correlations with the same coefficients as cases 17, 24, and 25. Case 28 ran with no temporal correlations ($\tau_N = .01$) but again with the same spatial correlations as cases 27 and 28. Case 29 was the reverse of case 28; no spatial correlations but high temporal correlations were used (.95 for one second). The results for all cases are shown in Table XIII. Case 26 portrays a target against a background with considerable spatial and time correlated infrared structure. Filter performance was poor with loss of track resulting in 12 simulations. Case 27 is the same as case 26 but with less time correlation in the background structure. Performance was similarly poor. Case 28 emulates a background that is correlated in space but not in time; performance, while not acceptable, was very close to that of the baseline, case 25. Case 29 shows a background that is highly time correlated but is not related

TABLE XIII

Temporal and Spatial Background Noise in the Truth
Model (S/N = 2): Case 26, 27, 28 and 29 Results

	Case 26		Case 27	
	.95 Time Cor.	Spatial Cor.	.5 Time Cor.	Spatial Cor.
	x	y	x	y
Average Standard Deviation				
Atmospheric Jitter				
Actual	.22	.20	.20	.20
Filter-Indicated	.19	.19	.19	.19
Target Position				
Actual	19.0	12.0	12.0	9.0
Filter-Indicated	.34	.34	.34	.34
Average Absolute Error				
Atmospheric Jitter	.06	.05	.05	.05
Target Position	1.0	1.0	1.0	1.0
Loss of Track	12 times		12 times	
	Case 28		Case 29	
	0 Time Cor.	Spatial Cor.	.95 Time Cor.	No Spatial Cor.
	x	y	x	y
Average Standard Deviation				
Atmospheric Jitter				
Actual	.20	.20	.19	.19
Filter-Indicated	.19	.19	.19	.19
Target Position				
Actual	2.0	1.6	4.0	1.6
Filter-Indicated	.34	.34	.34	.34
Average Absolute Error				
Atmospheric Jitter	.05	.05	.05	.05
Target Position	1.0	1.0	1.0	.50
Loss of Track	1 time		8 times	

spatially (an unrealistic background). Filter performance is poor but better than that displayed by cases 26 and 27 and worse than the performance of case 28. In summary, at low S/N temporally and spatially correlated background noise can significantly hinder filter performance. These cases suggest that the temporal noise correlations may be more critical than the spatial noise correlations. This may not be true in reality, because the spatial correlations were fairly realistic (see Appendix B) whereas the temporal correlations were worst case and possibly far from reality (see Appendix C). Target motion will also tend to decrease the importance of temporal correlations.

The last two cases tested were run with S/N equal to 2 with the target moving. The motion scenario previously described in this chapter and Chapter II was diagonal up and to the right with an initial velocity of 28.2 pixels per second. Case 30 was run without temporal and spatial background correlations; case 31 was run with them. The results are shown in Table XIV. For both cases the combination of target motion with low S/N resulted in almost equal, unacceptable filter performance. The cases provided little new insight to the filter. With motions greater than eight pixels per sample period which is very easily reached for most air-to-air missile scenarios, temporal modelling of the background changes completely in a single sample period.

Major Problem Areas

The results of the computer simulations presented in this chapter have provided considerable insight to the weakness of the original filter. From the parameter mismatch runs the following list of major problem areas in order of importance resulted:

- Low signal to noise ratio

- Target size mismatch

TABLE XIV

Temporal and Spatial Background Noise with Target Motion ($\dot{x}_0 = \dot{y}_0 = 20$ pixels/sec) in the Truth Model S/N = 2 Case 30 and 31 Results

	Case 30		Case 31	
	Baseline			
	0 Time Cor. No Spatial Cor.		.95 Time Cor. Spatial Cor.	
	x	y	x	y
Average Standard Deviation				
Atmospheric Jitter				
Actual	.19	.19	.19	.19
Filter-Indicated	.19	.19	.19	.19
Target Position				
Actual	7.0	10.0	10.0	10.0
Filter-Indicated	.34	.34	.34	.34
Average Absolute Error				
Atmospheric Jitter	.08	.05	.06	.08
Target Position	7.0	10.0	6.0	12.0
Loss of Track	17 times		18 times	

3. Target intensity mismatch (very much related to the low S/N problems)
4. The ratio of rms target dynamics to rms atmospheric jitter mismatch.

From the new truth model cases another list of significant problems are noted as follows (in decreasing order of concern):

1. Target motion
2. Low S/N ratio
3. Target shape and orientation mismatch
4. Temporal and spatial background noise at low S/N (but at moderate expected S/N levels these effects were not as significant as others).

The various solutions to these problems range from simple to very complex. The low signal to noise problem has been noted in many systems dealing with electromagnetic radiation. It was felt that the scope of this research should be limited to exclude this major problem area. The problem areas in this research which appeared significant and could be addressed within the allotted time period were target motion, target size and shape, and target intensity. The solution of the target intensity problems will help solve the S/N ratio problems noted in this chapter. Possible filter changes to maintain track on a moving target cover a large range of models and techniques, both adaptive and ad hoc, and will encompass much of the rest of this research. For the problems caused by target size, shape, and intensity, the apparent solution is real time adaptive estimation of the target parameters. Adaptive estimation schemes for estimating these parameters range from such complicated formulations as the solution of the full maximum likelihood equations to simple approximations and ad hoc schemes (Ref 16:10). These techniques will be studied in this research, attempting to achieve a final design with desirable performance and yet online applicability.

IV. Filter Requirements

Introduction

Section III was designed to indicate the changes needed to Capt. Mercier's filter to enable it to track near range moving targets. Several areas needing attention are as follows: (1) The filter must have the ability to predict future (next sample time) position, as the missile can travel many times the width of the FOV in one sample period. At 10 km a typical missile can travel approximately 80 pixels per sample period. With an 8 x 8 FOV and no prediction capability, the target image will depart the FOV, resulting in loss of track. Therefore, the tracker must have a predicted position for the next sample time. (2) The filter must be able to estimate orientation, size and shape of the image. This is necessary because no a priori estimate of flight path is possible and also missiles differ in length and aspect ratio (length:diameter). (3) The filter must determine the extent to which a particular flight path differs from the dynamics model in the filter, in other words, establish (i.e. adaptively estimate) the strength of the driving noise for its state stochastic process model. Flight path identification can be viewed as essentially maneuver detection followed by adjustment of the noise strength to fit the trajectory, and alternative means of responding to detected maneuvers should be considered. (4) The filter must have a good estimate of the maximum intensity of the image. As the residuals are the key to any parameter identification technique, a large difference between the filter's assumed I_{\max} (I_F) and the true I_{\max} will yield residuals with inaccurate information. Each of the above mentioned requirements and the basic Kalman filter equations will be discussed individually in the following paragraphs.

For the filter to predict future position, a current estimate of velocity is required as a minimum. Using the

predicted position, the controller was assumed to be able to point without error. Actually, a filter without prediction capability could track a moving target if it did not leave the FOV in a sample period. However, the pointing command would still be one sample behind at all times. At best, without predicting future position, the tracker would lag the missile considerably and in many cases lose it entirely.

The six state filter will be described first and in some detail. The eight state filter will be presented briefly showing the differences between the two filters.

Six State Filter

The dynamic model for the six state filter is

$$\dot{\underline{x}}_D(t) = \underline{v}_D(t) \quad (94)$$

$$\dot{\underline{v}}_D(t) = \underline{w}_D(t) \quad (95)$$

where the subscript D stands for dynamics, and

$$E[\underline{w}_D(t)] = \underline{0} \quad (96)$$

$$E[\underline{w}_D(t)\underline{w}_D^T(s)] = \sigma_D^2 \delta(t-s) \quad (97)$$

Equations (94) and (95) above are not strictly correct, as the tracker orientation is fixed with the LOS which is a rotating reference frame. However, a sample period of 1/30 second was considered short compared to any time constant describing missile acceleration. In other words, inertial equations were considered adequate for one sample period. To correct the filter model, equations similar to (63) and (65) in the truth model section, or, at least Coriolis theorem terms would need to be incorporated. Explicit correction of the rotating frame effects would only be incorporated if warranted in a performance/computational loading tradeoff.

The atmospheric disturbance model for each axis in the filter is

$$\dot{x}_A(t) = \frac{-1}{\tau_A} x_A(t) + w_A(t) \quad (98)$$

where the subscript A stands for atmospherics, and

τ_A = correlation time

$$E[w_A(t)] = 0 \quad (99)$$

$$E[w_A(t)w_A(s)] = \frac{2\sigma_A^2}{\tau_A} \delta(t-s) \quad (100)$$

For a detailed explanation of this model see Ref 5 and Appendix A.

Combining the dynamic and atmospheric models yields a propagation equation upon which to base a Kalman filter of

$$\dot{\underline{x}}_F = \underline{F}(t)\underline{x}_F(t) + \underline{G}_F(t)\underline{w}_F(t) \quad (101)$$

where the subscript F stands for filter, and

$\underline{F}(t) = \underline{F}$ = filter plant matrix

$\underline{x}_F(t)$ = filter state vector

x_1 = azimuth position

x_2 = elevation position

x_3 = azimuth velocity = \dot{x}_1

x_4 = elevation velocity = \dot{x}_2

x_5 = azimuth atmospheric disturbance

x_6 = elevation atmospheric disturbance

$\underline{G}_F(t) = \underline{G}_F$ = 6 x 4 noise distribution matrix

$E[\underline{w}_F(t)] = 0$

$$E[\underline{w}_F(t)\underline{w}_F^T(s)] = \underline{Q}_F \delta(t-s)$$

$$= \begin{bmatrix} \sigma_D^2 & 0 & 0 & 0 \\ 0 & \sigma_D^2 & 0 & 0 \\ 0 & 0 & \frac{2\sigma_A^2}{\tau_A} & 0 \\ 0 & 0 & 0 & \frac{2\sigma_A^2}{\tau_A} \end{bmatrix} \delta(t-s) \quad (102)$$

σ_D^2 = strength of velocity noise

σ_A^2 = variance of atmospheric jitter
model output

The velocity noise strength, σ_D^2 , need not be equal for the azimuth and elevation axes, conceptually. For this research, the axis noise strengths were set equal for the nonadaptive phases of tracking (acquisition) and allowed to be unequal during adaptive phases of tracking.

Explicitly, equation (101) is

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{\tau_A} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_A} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \quad (103)$$

where w_1 through w_4 are independent white Gaussian noises. The filter state and covariance propagation equations that result from this propagation equation are

$$\hat{\underline{x}}(t_{i+1}^-) = \underline{\phi}_F(t_{i+1}, t_i) \hat{\underline{x}}(t_i^+) \quad (104)$$

with covariance of

$$\begin{aligned} \underline{P}(t_{i+1}^-) &= \underline{\Phi}_F(t_{i+1}, t_i) \underline{P}(t_i^+) \underline{\Phi}_F^T(t_{i+1}, t_i) + \\ &\int_{t_i}^{t_{i+1}} \underline{\Phi}_F(t_{i+1}, \tau) \underline{G}_F(\tau) \underline{Q}_F(\tau) \underline{G}_F^T(\tau) \underline{\Phi}_F^T(t_{i+1}, \tau) d\tau \end{aligned} \quad (105)$$

The - and + superscripts refer to before and after a measurement update is made. Continuing, the state transition matrix is

$$\underline{\Phi}_F = \begin{bmatrix} 1 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{-\frac{\Delta t}{\tau_A}} & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-\frac{\Delta t}{\tau_A}} \end{bmatrix} \quad (106)$$

Note that $\underline{\Phi}_F$ is a constant matrix. The integral term in equation (105) is evaluated and becomes a representation of the growth in uncertainty due to driving noise since the last measurement update. This integral, also a constant denoted as \underline{Q}_{Fd} , is

$$\underline{Q}_{Fd} = \begin{bmatrix} \frac{\Delta t^3 \sigma_D^2}{3} & 0 & \frac{\Delta t^2 \sigma_D^2}{2} & 0 & 0 & 0 \\ 0 & \frac{\Delta t^3 \sigma_D^2}{3} & 0 & \frac{\Delta t^2 \sigma_D^2}{2} & 0 & 0 \\ \frac{\Delta t^2 \sigma_D^2}{2} & 0 & \Delta t \sigma_D^2 & 0 & 0 & 0 \\ 0 & \frac{\Delta t^2 \sigma_D^2}{2} & 0 & \Delta t \sigma_D^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_A^2 (1 - e^{-\frac{2\Delta t}{\tau_A}}) & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_A^2 (1 - e^{-\frac{2\Delta t}{\tau_A}}) \end{bmatrix} \quad (107)$$

so the covariance propagation equation becomes

$$\underline{P}(t_{i+1}^-) = \underline{\Phi}_F \underline{P}(t_i^+) \underline{\Phi}_F^T + \underline{Q}_{Fd} \quad (108)$$

The above derivation can be verified with any text on Kalman filtering (Ref 8:171).

Eight State Filter

The six state filter above is, to be sure, not the only possibility for filter formulation. An eight state filter including acceleration in both directions would seem to be appropriate also. For eight states the dynamic model would be, for example

$$\dot{\underline{x}}_D = \underline{v}_D \quad (109)$$

$$\dot{\underline{v}}_D = \underline{a}_D \quad (110)$$

$$\dot{\underline{a}}_D = \underline{w}_D \quad (111)$$

where \underline{w}_D is a vector of zero mean white Gaussian noise. Another model which allows for shorter time correlation of the acceleration is

$$\dot{\underline{a}} = \frac{-1}{\tau} \underline{a} + \underline{w}_D \quad (112)$$

where \underline{w}_D is a vector of zero mean white Gaussian noise and τ is the correlation time. This model will give more rapid variations in \underline{a} than equation (111) but introduces another parameter, τ , to estimate.

A third possible model for \underline{a} is

$$\dot{\underline{a}} = -\omega^2 \underline{v} + \underline{w}_D \quad (113)$$

where

$$\underline{v} = \text{velocity}$$

$$\omega = \frac{|\underline{v} \times \underline{a}|}{|\underline{v}|^2} \quad (114)$$

and \underline{w}_D is a vector of zero mean white Gaussian noise. This model is more complex and computationally less attractive, but it forms the basis of a model suggested for enhancing close range tracking (Ref 17).

Consideration was given to these models and equation (111) was chosen. Equation (114) is better suited to an aircraft rather than a missile. Combined with increased computation load, it seemed less appropriate than the other models. With adaptive estimation of the driving noise, equation (111) would give adequate performance in low and high acceleration cases without the need for equation (112).

Some advantages of the eight state filter are better characterization of the true acceleration than the previous (six state) model and better prediction capability. The price paid for this is longer computation time and more memory. Other advantages, associated with noninertial acceleration compensation and maneuver detection, will be discussed subsequently.

The statistics of the dynamic driving noise are

$$E[\underline{w}_D(t)] = \underline{0} \quad (115)$$

$$E[\underline{w}_D(t)\underline{w}_D^T(s)] = \sigma_D^2 \underline{I} \delta(t-s) \quad (116)$$

where the noise now enters at the jerk (time derivative of acceleration) level. Again, this model is not entirely correct either, but is a better approximation than that of the six state filter. The development of the eight state filter parallels the development of the six state filter so only the results will be presented here. The state vector is

x_1 = azimuth position

x_2 = elevation position

x_3 = azimuth velocity = \dot{x}_1

x_4 = elevation velocity = \dot{x}_2

x_5 = azimuth acceleration = \dot{x}_3

x_6 = elevation acceleration = \dot{x}_4

x_7 = azimuth atmospheric disturbance

x_8 = elevation atmospheric disturbance

$$\underline{\Phi}_F = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{\Delta t}{\tau_A}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{\Delta t}{\tau_A}} \end{bmatrix} \quad (117)$$

and

$$\underline{Q}_{Fd} = \begin{bmatrix} \frac{\Delta t^5 \sigma_D^2}{20} & 0 & \frac{\Delta t^4 \sigma_D^2}{8} & 0 & \frac{\Delta t^3 \sigma_D^2}{6} & 0 & 0 & 0 \\ 0 & \frac{\Delta t^5 \sigma_D^2}{20} & 0 & \frac{\Delta t^4 \sigma_D^2}{8} & 0 & \frac{\Delta t^3 \sigma_D^2}{6} & 0 & 0 \\ \frac{\Delta t^4 \sigma_D^2}{8} & 0 & \frac{\Delta t^3 \sigma_D^2}{3} & 0 & \frac{\Delta t^2 \sigma_D^2}{2} & 0 & 0 & 0 \\ 0 & \frac{\Delta t^4 \sigma_D^2}{8} & 0 & \frac{\Delta t^3 \sigma_D^2}{3} & 0 & \frac{\Delta t^2 \sigma_D^2}{2} & 0 & 0 \\ \frac{\Delta t^3 \sigma_D^2}{6} & 0 & \frac{\Delta t^2 \sigma_D^2}{2} & 0 & \Delta t \sigma_D^2 & 0 & 0 & 0 \\ 0 & \frac{\Delta t^3 \sigma_D^2}{6} & 0 & \frac{\Delta t^2 \sigma_D^2}{2} & 0 & \Delta t \sigma_D^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_A^2 (1 - e^{-\frac{2\Delta t}{\tau_A}}) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_A^2 (1 - e^{-\frac{2\Delta t}{\tau_A}}) \end{bmatrix} \quad (118)$$

The advantages and disadvantages of both six and eight state formulations were considered and, unless performance required the eight state filter to be used, the six state filter would be pursued. Thus, the six state filter above is the basic formulation. The adaptive estimation of the \underline{Q}_{Fd} matrix and other modifications of this basic structure will be discussed later in this section.

Size and Shape Estimation

The second major requirement is to determine the size, shape, and orientation of the image ellipse. Figure 19 shows possible variables to describe the image. The subscripts v and pv refer to the image plane velocity direction and perpendicular to that velocity direction, respectively.

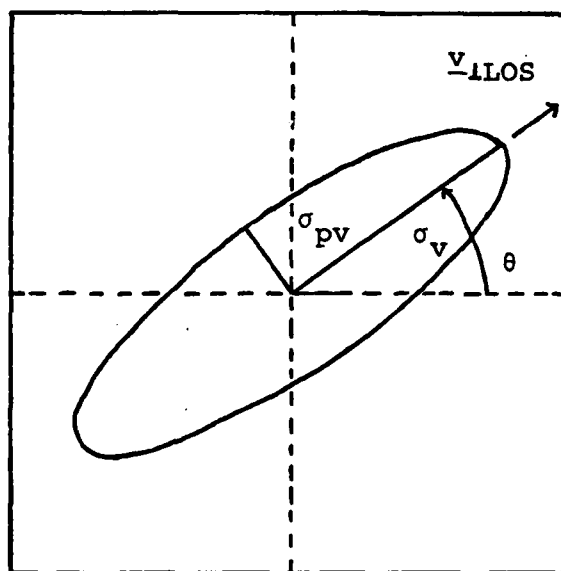


Figure 19. Image Geometry Characteristics

Define

$$AR = \text{aspect ratio} = \frac{\sigma_v}{\sigma_{pv}} \quad (119)$$

The measurement update for this Kalman filter is so intimately related to the estimation of σ_v , σ_{pv} and/or AR that they will be treated together.

The assumed measurement relation for an Extended Kalman Filter (EKF) is

$$\underline{z}(t_i) = \underline{h}(\underline{x}(t_i), t_i) + \underline{v}(t_i) \quad (120)$$

where

$$E[\underline{v}(t_i)] = \underline{0} \quad (121)$$

$$E[\underline{v}(t_i)\underline{v}^T(t_j)] = \underline{R}(t_i)\delta_{ij} \quad (122)$$

and \underline{h} is a nonlinear function relating the states to the measurements. Specifically, for the j th of 64 pixels

$$\begin{aligned} & h_j(\underline{x}(t_i), t_i) \\ &= \frac{1}{A_{pj}} \iint_{A_{pj}} I_F(t_i) \exp \left\{ -\frac{1}{2} [\Delta x_v \Delta y_v] \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_{pv}^2 \end{bmatrix}^{-1} [\Delta x_v \Delta y_v]^T \right\} dA_{pj} \end{aligned} \quad (123)$$

where

A_{pj} = area of pixel j , $j=1, \dots, 64$

I_F = (filter estimate of) maximum intensity

Δx_v , Δy_v have similar meaning as in eqns. (30) and (31), and are measured with respect to the principal axes of the ellipse in Figure 19.

σ_v = (filter estimate of) semimajor axis of ellipse in Figure 19

σ_{pv} = (filter estimate of) semiminor axis of ellipse in Figure 19

Equation (123) is evaluated like equation (7) in the truth model section except that instead of averaging evaluations of the integrand over 16 interior points of each pixel,

only the midpoint is used. This is done to reduce the computational loading in the filter. Also, since the angle θ in Figure 6 is not available to the filter, it is never computed. Instead $\cos\theta$ and $\sin\theta$ are estimated from \hat{x}_3 (azimuth velocity) and \hat{x}_4 (elevation velocity) under the assumption that the major axis of the ellipse is colinear with the image plane velocity (\underline{v}_{LOS} of Figure 7).

The measurement update relations for an Extended Kalman Filter using the inverse covariance form are

$$\underline{P}^{-1}(t_i^+) = \underline{P}^{-1}(t_i^-) + \underline{H}^T(t_i) \underline{R}^{-1}(t_i) \underline{H}(t_i) \quad (124)$$

$$\underline{P}(t_i^+) = [\underline{P}^{-1}(t_i^+)]^{-1} \quad (125)$$

$$\underline{K}(t_i) = \underline{P}(t_i^+) \underline{H}^T(t_i) \underline{R}^{-1}(t_i) \quad (126)$$

$$\hat{\underline{x}}(t_i^+) = \hat{\underline{x}}(t_i^-) + \underline{K}(t_i) \{ \underline{\Gamma}(t_i) - \underline{h}(\hat{\underline{x}}(t_i^-), t_i) \} \quad (127)$$

where $\underline{\Gamma}(t_i)$ is a sample of \underline{z} at time t_i .

The \underline{R} matrix is the covariance of the measurement model. For pixels with assumed independent, equal variance noise, \underline{R} will be constant in time and of the form

$$\underline{R} = R_F \underline{I} \quad \text{and} \quad \underline{R}^{-1} = \frac{1}{R_F} \underline{I} \quad (128)$$

The \underline{H} matrix is defined as

$$\underline{H}(t_i) = \left. \frac{\partial \underline{h}(\underline{x}, t)}{\partial \underline{x}} \right|_{\underline{x} = \hat{\underline{x}}(t_i^-)} \quad (129)$$

For details of how this derivative is computed, see Ref 5 and the program listing (subroutine MEASF) found in Appendix F.

The inverse covariance form of the measurement update equations was used for computational efficiency. This formulation requires inversions of two matrices both of the

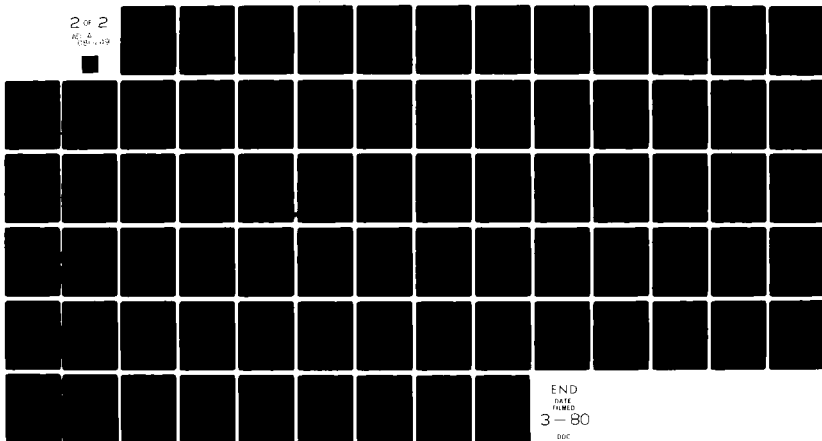
AD-A088 249

AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/6 17/5
AN ADAPTIVE DISTRIBUTED-MEASUREMENT EXTENDED KALMAN FILTER FOR --ETC(U)
DEC 79 R L JENSEN, D A HARNLY
AFIT/6A/EE/79-1-VOL-1

UNCLASSIFIED

NL

2 of 2
AD-A
108 108



END
DATE
FILMED
3-80

DOC

state vector dimension. Had the standard Kalman filter equations been used, only one inversion would have been required. But, that one inversion would be for a square matrix of the order of the measurement vector dimension, a 64 x 64 matrix.

Returning to the problem of identifying σ_v , σ_{pv} and/or AR, there are several methods available. The uncertain parameters can be included in the state vector and, if modelled as Brownian (as in this research), requires statistical knowledge of the uncertain parameters, which is often difficult to provide. Also, the appropriate driving noise strength for the dynamics model may be difficult to interpret physically.

Complete maximum likelihood equations generally yield good results but could never be implemented on line (Ref 16:10-126). Multiple model adaptive estimation (Ref 16:10-103) is a possibility, as are some ad hoc procedures (Ref 16:10-121). Multiple models, however, would require many models to span the variation of missile size with range and projected view with respect to LOS. The method finally chosen was a gradient minimization technique. It approximates full scale Maximum Likelihood estimation of states and parameters, but with significant computational advantage, critical to online implementation (Ref 10:10-1).

Equation (123), the filter measurement relation, can be rewritten with the parameters of interest in evaluated form as

$$\begin{aligned} \underline{h}(\underline{x}(t_i), \sigma_v, AR, \theta, t_i) = I_F \exp \left\{ -\frac{1}{2} \left[\frac{((x-x_p) \cos \theta + (y-y_p) \sin \theta)^2}{\sigma_v^2} \right] \right. \\ \left. + \left[\frac{((y-y_p) \cos \theta - (x-x_p) \sin \theta)^2}{\sigma_v^2} AR^2 \right] \right\} \end{aligned} \quad (130)$$

Note that only two of σ_v , σ_{pv} or AR need be estimated as they are related by equation (119). For simulation purposes, AR and σ_v were estimated so as to be directly compared to similar parameters already in the truth model. For actual implementation, a tradeoff analysis between performance and efficiency would have to be made to determine which of the three possibilities (σ_v and AR, σ_v and σ_{pv} , σ_{pv} and AR) is best.

The approach used to estimate σ_v and AR was a gradient recursion, minimizing the quadratic cost function given below in equation (131).

$$L(\underline{x}(t_i), \sigma_v, AR, \theta, t_i) \\ = [\underline{\Gamma}(t_i) - \underline{h}(\hat{\underline{x}}(t_i^-), t_i)]^T [\underline{\Gamma}(t_i) - \underline{h}(\hat{\underline{x}}(t_i^-), t_i)] \quad (131)$$

This equation can be viewed as a simplified form of maximizing

$$L'(\underline{\Gamma}(t_i), \underline{A}) = \ln f_{\underline{z}(t_i) | \underline{z}(t_{i-1}), \underline{A}}(\underline{\Gamma}(t_i) | \underline{z}(t_{i-1}), \underline{b}) \quad (131a)$$

where f is the density function of the current measurement based on all previous measurements and the parameter vector (\underline{A}). Explicitly equation (131a) is

$$L'(\underline{\Gamma}(t_i), \underline{A}) = \ln[2\pi^{-m/2} |\underline{P}_L(t_i)|^{-1/2}] \\ - \frac{1}{2} [\underline{\Gamma}(t_i) - \underline{h}(\hat{\underline{x}}(t_i^-), t_i)]^T \underline{P}_L^{-1}(t_i) [\underline{\Gamma}(t_i) - \underline{h}(\hat{\underline{x}}(t_i^-), t_i)] \quad (131b)$$

$$\text{where } \underline{P}_L(t_i) = \underline{H}(t_i) \underline{P}(t_i^-) \underline{H}^T(t_i) + \underline{R}(t_i)$$

The conditional density function of equation (131a) will be maximized when equation (131b) is maximized. The first term in (131b) does not affect the maximization and in the second term \underline{P}_L^{-1} is deleted primarily because of low confidence in

it. Also, having 64 measurements may tend to smooth the function. Removing P_L^{-1} was assumed not to change the position of the minimum substantially. Then the estimator equations are

$$\hat{AR}_F(t_i) = \hat{AR}_F(t_{i-1}) + c \left. \frac{\partial L}{\partial AR} \right|_{AR = \hat{AR}_F} \quad (132)$$

$$\hat{\sigma}_{VF}(t_i) = \hat{\sigma}_{VF}(t_{i-1}) + c \left. \frac{\partial L}{\partial \sigma_V} \right|_{\sigma_V = \hat{\sigma}_{VF}} \quad (133)$$

where

c = a constant determined empirically which is the same in (132) and (133)

Another advantage of this method is that part of the computation needed for the partial derivatives in equations (132) and (133) is already done in evaluation of equation (129). All three partial derivatives ($\frac{\partial h}{\partial x}$, $\frac{\partial L}{\partial AR}$, $\frac{\partial L}{\partial \sigma_V}$) are computed in the MEASF subroutine. See the program listing for details of equations (132) and (133).

Although AR and σ_V were estimated every sample period here, this may be avoided. AR and σ_V do not change rapidly and could be estimated at a slower rate, for example every other sample or every five samples. One caution should be noted, however. Other portions of the filter algorithm estimate various parameters and these estimations are also dependent on the residuals given by

$$\underline{r}(t_i) = \underline{I}(t_i) - \underline{h}(\underline{x}(t_i^-), t_i) \quad (134)$$

The residuals are in turn dependent on $\underline{h}(\underline{x}(t_i^-), t_i)$ and therefore AR and σ_V . So, equations (132) and (133) must acquire correct values rapidly and maintain them. If AR and σ_V estimates differ considerably from the true values, the residuals may be in error, making the estimation of other uncertain parameters difficult.

Dynamic Driving Noise Estimation

The performance of any Kalman filter depends on adequate knowledge of certain characteristics. Some of these that must be known are the statistics describing the dynamic driving noise. Several methods to estimate the driving noise adaptively are maximum likelihood, multiple model adaptive (Bayesian) techniques, correlation, and covariance matching techniques (Ref 16:Chapter 10). The ability to change \underline{Q}_{Fd} in real time was deemed necessary because an air-to-air missile can exhibit a wide range of dynamic characteristics. Therefore, maximum tracking accuracy could not be achieved with a single choice of \underline{Q}_{Fd} . Even with adaptive estimation of \underline{Q}_{Fd} , a compromise between statistically accurate estimates and rapid response to a change in \underline{Q}_{Fd} is required. The two concepts tend to work against each other and it is therefore difficult to design an estimator with both attributes.

Associated with an adaptive \underline{Q}_{Fd} estimation, there seems to be two logical divisions of state estimation, an acquisition mode and a tracking mode. To start the filter algorithm, an estimate of the state vector and covariance must be provided. Because of the distances an air-to-air missile can travel in one sample time (80 pixels or so) good estimates of position and velocity are required by the filter. To allow for errors in values handed off to the filter, from a radar for example, an acquisition mode was used.

The acquisition mode consisted of an initial filter state covariance matrix \underline{P}_0 and driving noise \underline{Q}_{Fd} , with significantly increased values over what normally would be used for the tracking mode. The \underline{P}_0 matrix would be a function of the error statistics of the hand off device and \underline{Q}_{Fd} would be a function of known deficiencies in the dynamic model of the filter. \underline{P} was then allowed to propagate and update normally, using \underline{Q}_{Fd} values appropriate for tracking, thereby reducing the covariance matrix to its proper values.

The \underline{Q}_{Fd} matrix in equation (107) is the result of a stationary process and is normally a constant matrix. To obtain adequate performance for a maneuvering target \underline{Q}_{Fd} needs to be "larger" than that necessary for a benign target. This larger value for \underline{Q}_{Fd} will, however, degrade the filter performance with the benign target. For the best performance possible in both situations, \underline{Q}_{Fd} needs to be estimated adaptively. Adaptive estimation was not done during the acquisition phase (first .5 sec) unless the noise strength that would represent the maneuver exceeded the filter noise strength. Instead σ_D^2 in equation (107) started at a value larger than normal tracking would require (typically $\sigma_D^2 = 600 \text{ pixels/sec}^3$ for the six state filter or 600 pixels/sec^5 for the eight state filter), then was reduced linearly to a value characteristic of a benign target, at .5 sec (typically $\sigma_D^2 = 150 \text{ pixels/sec}^3$ or pixels/sec^5 depending on the filter used). This reduces the number of parameters needing estimation immediately and precludes the possibility of induced stability problems. Also note that reducing σ_D^2 has no effect on the atmospheric portion of \underline{Q}_{Fd} during acquisition.

After the acquisition phase, adaptive estimation of \underline{Q}_{Fd} starts. Again, several methods can be used (Ref 16: Chapter 10). A full scale estimator based on maximum likelihood suffers from computational loading problems. This makes it virtually impossible to implement on line without simplifying assumptions. Multiple model adaptive estimation techniques are possible but would require more storage and computing capability. The estimator finally selected was an ad hoc procedure based on equation 10-93 of Ref 10. It can also be interpreted as an approximation to a maximum likelihood estimate.

Heuristically, the estimator can be developed as follows.

$$\underline{P}(t_i^-) = \underline{\Phi}_F(t_i, t_{i-1}) \underline{P}(t_{i-1}^+) \underline{\Phi}_F^T(t_i, t_{i-1}) + \underline{Q}_{Fd}(t_{i-1}) \quad (135)$$

also

$$\underline{P}(t_i^+) = \underline{P}(t_i^-) - \underline{K}(t_i) \underline{H}(t_i) \underline{P}(t_i^-) \quad (136)$$

solving equation (136) for $\underline{P}(t_i^-)$ yields

$$\underline{P}(t_i^-) = \underline{P}(t_i^+) + \underline{K}(t_i) \underline{H}(t_i) \underline{P}(t_i^-) \quad (137)$$

where the term $\underline{K}(t_i) \underline{H}(t_i) \underline{P}(t_i^-)$ is preserved intact because it will be estimated on the basis of observed residuals. Setting equation (135) equal to equation (137) and solving for \underline{Q}_{Fd} yields

$$\underline{Q}_{Fd} = \underline{K}(t_i) \underline{H}(t_i) \underline{P}(t_i^-) + \underline{P}(t_i^+) - \underline{\Phi}_F(t_i, t_{i-1}) \underline{P}(t_{i-1}^+) \underline{\Phi}_F^T(t_i, t_{i-1}) \quad (138)$$

Now all that is needed is an estimate of the first term in equation (138). The state update equation (127) can be written as

$$\hat{\underline{x}}(t_i^+) - \hat{\underline{x}}(t_i^-) = \underline{K}(t_i) r(t_i) = \Delta \underline{x}(t_i) \quad (139)$$

An estimate of $E[\Delta \underline{x}(t_i) \Delta \underline{x}^T(t_i)]$ can be had by forming

$$E[\Delta \underline{x}(t_i) \Delta \underline{x}^T(t_i)] \triangleq \frac{1}{N} \sum_{j=i-N+1}^i [\Delta \underline{x}(t_j) \Delta \underline{x}^T(t_j)] \quad (140)$$

Assuming the filter is operating in steady state, $\Delta \underline{x}(t_i)$ is a white sequence with covariance $\underline{K}(t_i) \underline{H}(t_i) \underline{P}(t_i^-)$ that is independent of $\underline{x}(t_i)$ (Ref 8:229).

$$\underline{K}(t_i) \underline{H}(t_i) \underline{P}(t_i^-) \cong \frac{1}{N} \sum_{j=i-N+1}^i [\Delta \underline{x}(t_j) \Delta \underline{x}^T(t_j)] \quad (141)$$

so an estimate of $\underline{Q}_{Fd}(t_i)$ becomes

$$\begin{aligned}\hat{\underline{Q}}_{Fd}(t_i) &= \frac{1}{N} \sum_{j=i-N+1}^i [\Delta \underline{x}(t_j) \Delta \underline{x}^T(t_j)] + \underline{P}(t_i^+) \\ &\quad - \underline{\phi}_F(t_i, t_{i-1}) \underline{P}(t_{i-1}^+) \underline{\phi}_F^T(t_i, t_{i-1})\end{aligned}\quad (142)$$

Rather than just average $\Delta \underline{x}(t_i) \Delta \underline{x}^T(t_i)$, the entire relation is averaged over the most recent N samples.

$$\begin{aligned}\hat{\underline{Q}}_{Fd}(t_i) &= \frac{1}{N} \sum_{j=i-N+1}^i \{ \Delta \underline{x}(t_j) \Delta \underline{x}^T(t_j) + \underline{P}(t_j^+) \\ &\quad - \underline{\phi}_F(t_j, t_{j-1}) \underline{P}(t_{j-1}^+) \underline{\phi}_F^T(t_j, t_{j-1}) \}\end{aligned}\quad (143)$$

One advantage of equation (143) is that all required quantities are readily available from the filter in the form needed. Also, to reduce storage requirements, a fading memory approximation to a finite memory was employed. That is

$$\hat{\underline{Q}}_{Fd}(t_i) = a \hat{\underline{Q}}_{Fd}(t_{i-1}) + (1-a) \underline{Q}_{Fdl}(t_i) \quad (144)$$

where $\underline{Q}_{Fdl}(t_i)$ is the single term in the summation of equation (142) corresponding to $j=i$ and a is a parameter which essentially controls how long old estimates of $\hat{\underline{Q}}_{Fd}$ are maintained. If $a=0$ this corresponds to using current \underline{Q}_{Fdl} only, while $a=1$ corresponds to ignoring current \underline{Q}_{Fdl} and using only old $\hat{\underline{Q}}_{Fd}$. Low a implies little confidence in previous estimates, while high a gives slow response to actual \underline{Q}_{Fd} changes.

Typical values of a are

$$.7 < a < 1 \quad (145)$$

From partial investigation of the sensitivity of $\hat{\underline{Q}}_{Fd}$ to this parameter and the desire for statistical confidence, a was set at .8 for this research.

I_{max} Estimation

Looking at equations (123) and (129), also seen in Chapter II, reveals that I_F plays an important role in providing accurate residuals. Because of both uncertainties and changes in I_{\max} with time, no a priori value is adequate for all probable cases. Therefore, on line estimation of I_F is required. For a detailed explanation of I_{\max} estimation see Chapter V.

Several methods are possible: (1) find the largest value in the measurement array and set I_F equal to it; (2) one dimensional curve fitting to several pixels adjacent to the pixel with the highest observed intensity and then setting I_F equal to the maximum of that curve; and (3) two dimensional surface fitting to several pixels surrounding the highest pixel and then setting I_F equal to the maximum of that surface. Method (1) was used here because it was easy to implement and provided adequate performance for initial investigations. In equation form the estimator is

$$I_F(t_j) = \max\{z_i(t_j) | 1 \leq i \leq 64\} \quad t = t_j \quad (146)$$

Improved estimation performance can be obtained via two modifications to this method.

Because I_{\max} does not vary rapidly (over a few samples), a smoother estimate can be obtained by averaging over time. This equation, similar to equation (144), is:

$$\hat{I}_F(t_j) = c \hat{I}_F(t_{j-1}) + (1-c)I_F(t_j) \quad (147)$$

where I_F is found by equation (146).

Even with time averaging this is still a biased estimate because $I_F(t_i)$ includes a noise component and Gaussian shape effects which will be explained further in Chapter V. An error function was determined empirically of the form

$$I_F(t_j) = I_F(t_j) + e(\sigma_v(t_j)) \quad (148)$$

This error function should depend upon I_F , σ_{pv} , x_{peak} , and y_{peak} also but they were not included because, for the tests using equation (146), I_{max} was not varied and the dependence on x_{peak} and y_{peak} can be neglected as long as $x_{peak} < 4$ and $y_{peak} < 4$ (intensity peak remains in FOV). Thus, the error function was chosen to be a function of only $\sigma_v(t_i)$. See the program listing and Chapter V for more details.

V. Adaptive Filter Performance and Modifications

Introduction

The basic new filter design described in Chapter IV is evaluated to determine its performance characteristics. Each facet of the adaptive filter, including the parameter estimation techniques, the acquisition scheme, the six state filter, and the eight state filter, is tested and its performance examined. First, the performance of the techniques for I_{\max} estimation are discussed. Then σ_v and AR estimates are compared to the truth model values; some differences are noted with possible explanations. Next, the acquisition strategy as formulated in Chapter IV is shown to be satisfactory for the trajectories studied and allows the filter to recover from limited hand-off errors. Errors caused by the non-inertial properties of the rotating LOS reference frame (non-inertial accelerations) are noted in cases in which the six state filter is used and are greatly removed by the use of the eight state filter. Designed originally to allow tracking of a maneuverable target, the Q_{Fd} estimation process exhibited stability problems for some steady state trajectories and continual loss of track for scenarios of extreme dynamic maneuver initiation. Modifications were attempted to detect vehicle maneuvers and enhance filter tracking. These modifications were tested and proved partially successful. (The input parameters and selected performance plots for all cases discussed in this chapter can be found in Appendices K and L, respectively.)

I_{\max} Estimation

The estimation of I_{\max} uses one or more of three techniques. The first technique as described in Chapter IV chooses I_F to be equal to the largest value in the measurement array for each sample period. The second technique, also described in Chapter IV, was the time averaging of the

maximum measurement as illustrated by equation (147). The third technique uses an error function to remove a bias caused by a combination of target size and shape and an upward shift of all simulated values which was purposely programmed to avoid negative measurement values. Physically, negative FLIR outputs are impossible assuming no instrument noise. So, after the measurements are generated in the truth model to reflect the target intensity function and after the simulated zero mean background noise is added to the measurements, all values are biased upwards so that the largest negative measurement is set to .1. This process actually models the physical effects of the time varying nature of the mean of the background noise. This last technique, which was briefly mentioned in Chapter IV, is developed in more detail in this section. Test data will demonstrate that a combination of the three techniques yields the best estimate of I_{\max} .

The maximum measurement technique was used in tracking a target, with $\sigma_v = 3$ pixels, $\sigma_{pv} = 1$ pixel and $I_{\max} = 25$ moving diagonally cross range, target trajectory 2 as described in Chapter II. The estimate for I_{\max} in this scenario is plotted in Figure 20. The data plotted is the I_{\max} estimate at every tenth of a second for the first three seconds of a single simulation run. The time average value of I_{\max} is 24.47, compared to a true value of 25, with a standard deviation of 1.12. This compares roughly to the standard deviation of the background noise, $\sigma_N = 1.414$.

Two characteristics of the distribution shown in Figure 20 are notable. First, the distribution of the points demonstrate that the I_{\max} estimates follow the added noise, resulting in an undesirably large dispersion of values. Second, the average I_{\max} estimate is less than the true value. The cause and solution for this trend will be discussed more in the two paragraphs to follow.

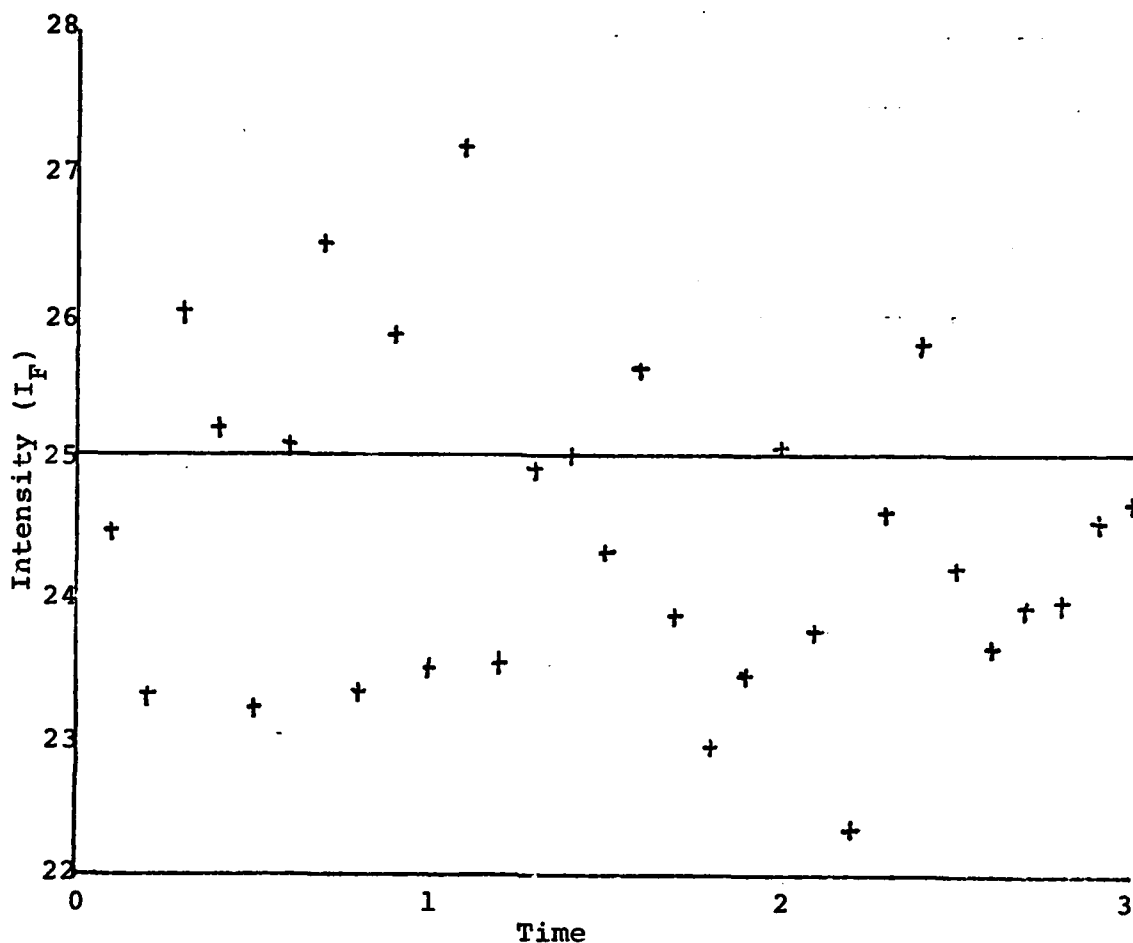


Figure 20. I_{\max} Estimate: Maximum Measurement

The second technique uses a time average of the maximum measurement value to determine an I_{\max} estimate. As shown in equation (147), the current maximum measurement is appropriately weighted and added to a weighted value of the previous estimate of I_{\max} . The purpose of this technique is to produce a smoother estimate of I_{\max} . Figure 21 shows that a better behaved trend is achieved. This graph uses the same data values as Figure 20, but time averaging is used weighting the current measurement maximum by .2 and the previous I_{\max} estimate by .8. The average I_{\max} is 24.53 with a standard deviation of .334. While the average is very close to the average arrived at without smoothing, the

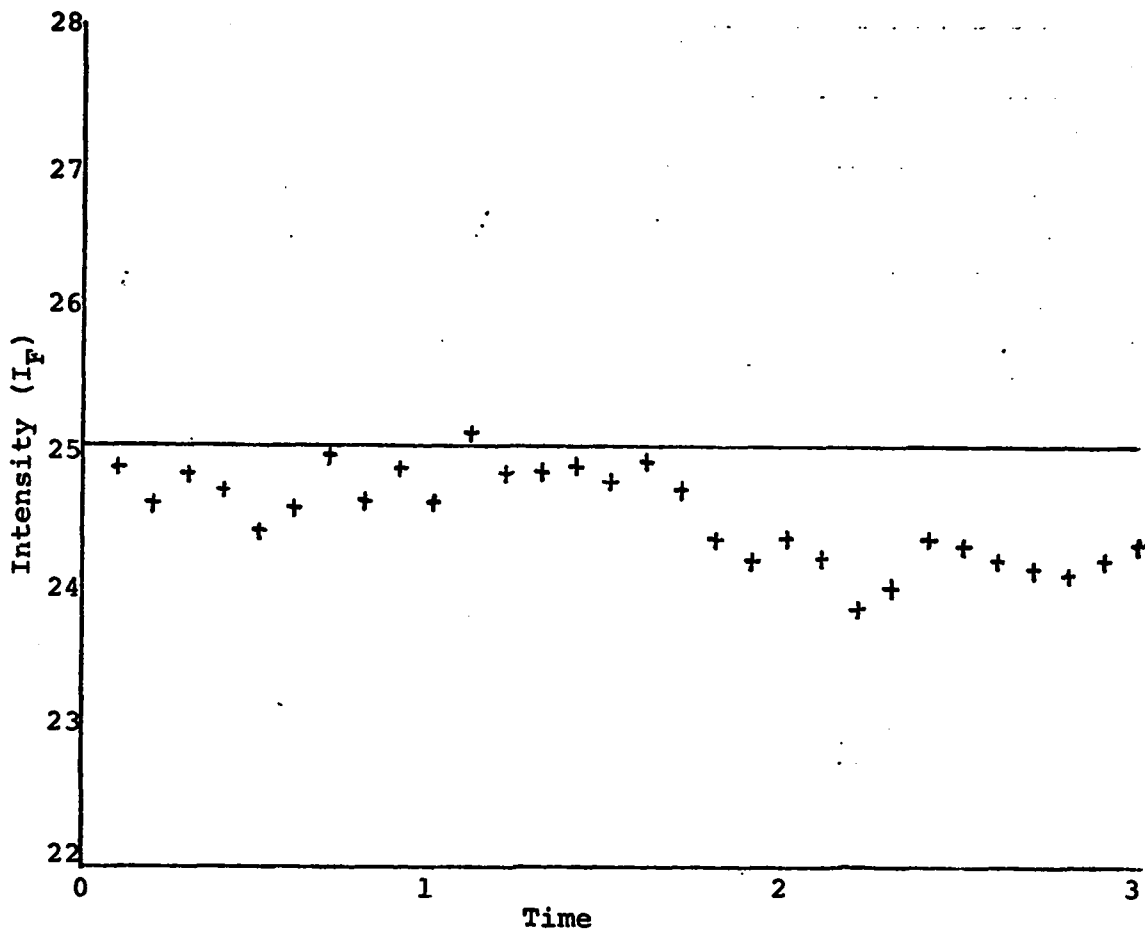


Figure 21. I_{\max} Estimate: Averaged Maximum Measurement

standard deviation has decreased by a factor of 3. Time averaging appears to be a very good technique to use.

For both of the techniques just discussed, the average estimated I_{\max} was about half of an intensity unit less than the true value. This fact is attributable to the summation of two effects. As previously mentioned, to avoid negative simulated measurement values (assumed not to be possible with the FLIR sensor as instrument noise is assumed negligible), all measurements are biased positively. This is explained by the truth model measurement scheme. After generating appropriate measurement values based on the targets

intensity function with I_{\max} equal to 25, background noise of known strength is added to each measurement. Where the target intensity is small, this process can result in negative values in the measurement array. To solve this problem all measurements are biased upward (positive) to make the largest negative value equal to .1.

The second effect is caused by a combination of target size and shape. Each measurement based on the Gaussian intensity function as defined in equation (7) represents the average target intensity over the area of the pixel. (The procedure for computing the pixel output due to target intensity in the truth model is more completely detailed in Appendix D.) Ignoring for a moment the added background noise, the average intensity value of a given pixel will be less than I_{\max} even if the maximum is found within the pixel. This is shown in one dimension in Figure 22. If the area under the Gaussian curve is integrated from 0 to 1 (i.e. one pixel) and then divided by the length of the interval, the resulting average, I_{avg} , is less than I_{\max} .

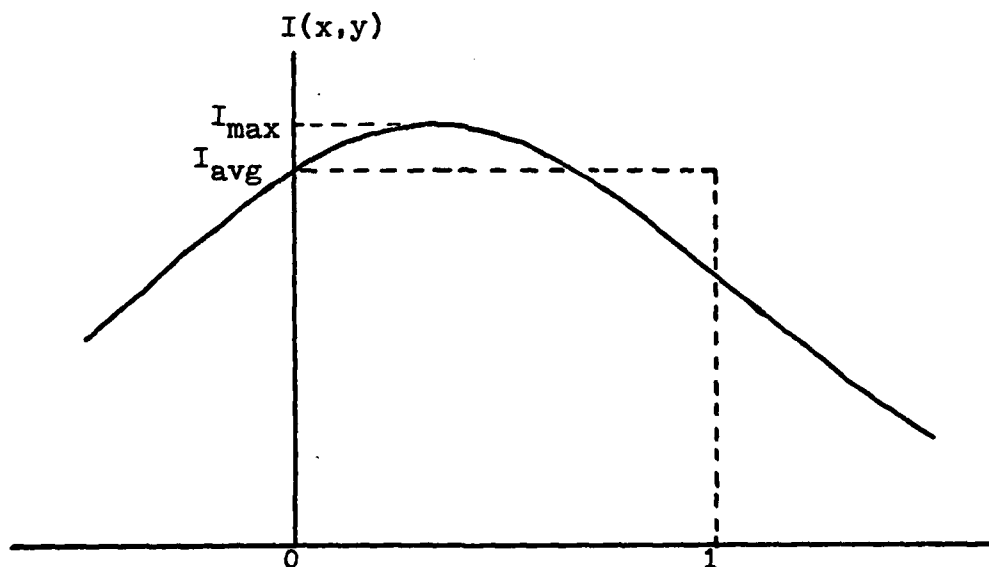


Figure 22. Average Value of Gaussian Curve Over Interval 0 to 1 (1 Pixel).

If the target is small in size, i.e. $\sigma_{pv} = 1$ pixel and $\sigma_v = 1$ pixel, I_{avg} can be significantly smaller than I_{max} because variation of the intensity function is large over one pixel. For a large target, the curvature of the function is more gradual and consequently the average intensity is closer to the maximum in the pixel. Conceptually, I_{avg} above can be expressed as a function of I_{max} , x_{peak} , y_{peak} , σ_v and σ_{pv} . This function could then be solved for I_{max} , but it would be a very complex process. Instead, this function can be approximated by a polynomial fit in terms of σ_v only, as described in the following paragraph.

The combination of the upward bias to avoid negative noise values and the downward bias due to target intensity function curvature resulted in an average I_{max} in Figures 20 and 21 of about 24.5 for $\sigma_{pv} = 1$ pixel and $\sigma_v = 3$ pixels. Varying the size of the target in the cross range trajectory results in different averages for I_{max} as shown in Table XV. In Table XV the subscripts F and T correspond to filter and truth model values, respectively. σ_{vF} is the estimated standard deviation of the target in the direction of the target's velocity vector. σ_v estimation and the biases noted between σ_{vT} and average σ_{vF} in Table XV are discussed more fully in the following section. The data used to establish Table XV values for $\sigma_{vT} = 3.0$ was not the same data as was used in Figures 20, 21, 23, and 24.

TABLE XV
 I_F Variation with σ_{vF} for Single Simulations
of Cross Range Target ($\sigma_{pv} = 1$ pixel)

σ_{vT}	Avg σ_{vF}	I_{max}	Avg I_F	Avg Error ($I_{max} - I_F$)
1.0	1.57	25.0	22.91	2.09
3.0	3.63	25.0	24.57	.43
5.0	4.78	25.0	25.00	.0

From the values of the errors and σ_{vF} , an error function can be derived to compute a correction for each I_{\max} estimate. A second order polynomial fit to the three points in Table XV ($\sigma_{vF} = 1.57$ pixels, error = 2.09; $\sigma_{vF} = 3.63$ pixels, error = .43; $\sigma_{vF} = 4.78$ pixels, error = 0) resulted in the following error function

$$e(\sigma_{vF}) = .136 \sigma_{vF}^2 - 1.518 \sigma_{vF} + 4.18 \quad (149)$$

A second order polynomial was chosen because it was simple to use and the coefficients were determined exactly by the three points. A more precise functional form could have been derived which would have required more data points and would have resulted in a more complex function. The I_{\max} estimates shown in Figure 20 were recomputed using the error function and are displayed in Figure 23. In Figure 23 the maximum measurement for each sample period is corrected with the error function using the filter's estimate of σ_{vF} for that sample period; time averaging is not used. The resulting statistics of the estimation process is an average I_F of 24.99 with a standard deviation of 1.16. The bias has been removed but the standard deviation is still large -- I_{\max} estimates are still reflecting the background noise. The error function was optimized for $I_{\max} = 25$, but would provide some correction if I_{\max} were other than 25. However, a time shortage did not allow a more detailed study. Further analysis could probably determine an approximating error function depending on σ_{vF} and $I_F(t_{i-1})$ (the previous estimate of I_{\max}) and other parameters also for an appropriate range of σ_v and I_{\max} .

The best I_{\max} estimation procedure incorporates the time averaging to smooth the trend in the estimates and the error function to correct for the bias in the average maximum measurement. Shown in Figure 24 are the same data points previously discussed with time averaging and error function. The average I_F is 24.99 with a standard deviation

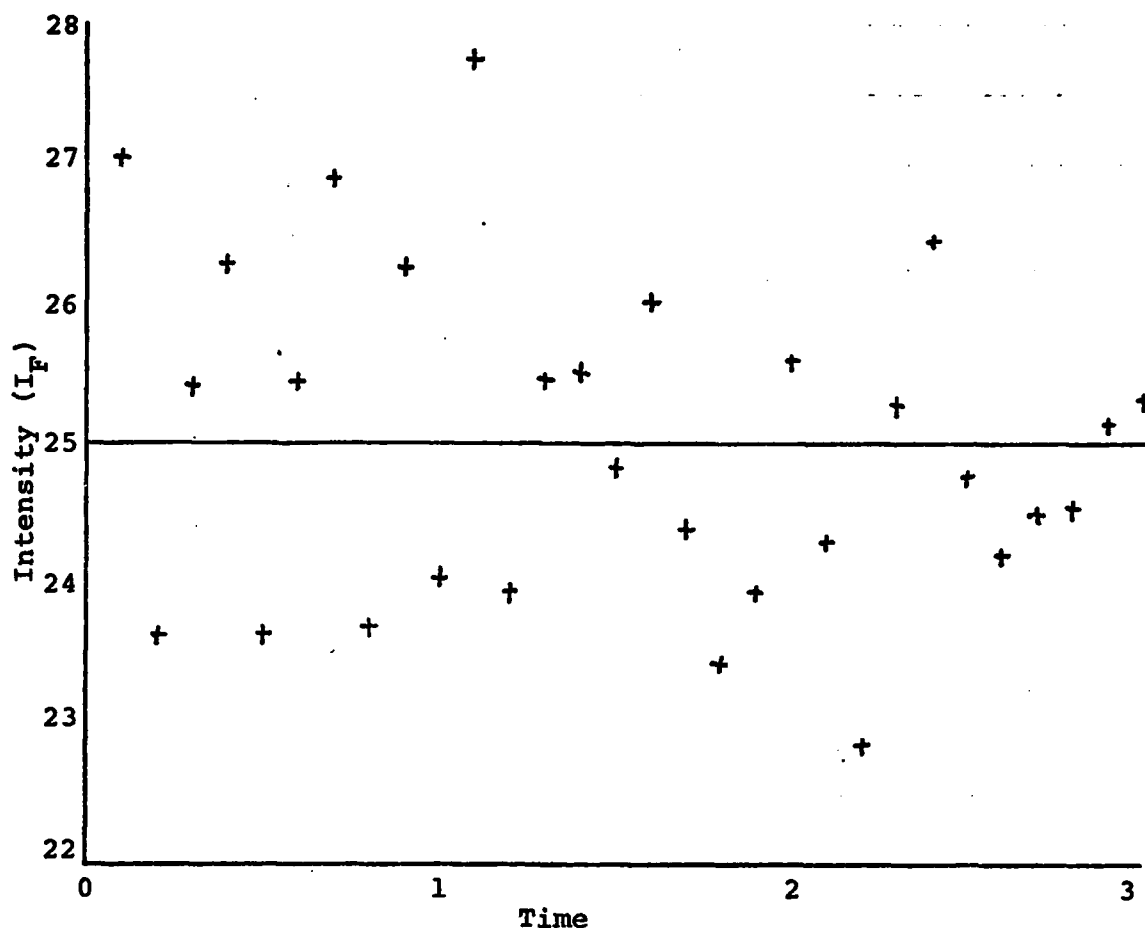


Figure 23. I_{\max} Estimate: Maximum Measurement With Error Function

of .299. These statistics and a visual examination of Figure 24 reveal the success of this combined estimation process. The bias is gone; the estimated I_{\max} values are grouped around the $I_F = 25$ line. Also, the distribution of points is well behaved, no longer tracking the total noise amplitude. This combined procedure results in an accurate I_{\max} estimate well worth the extra computations needed.

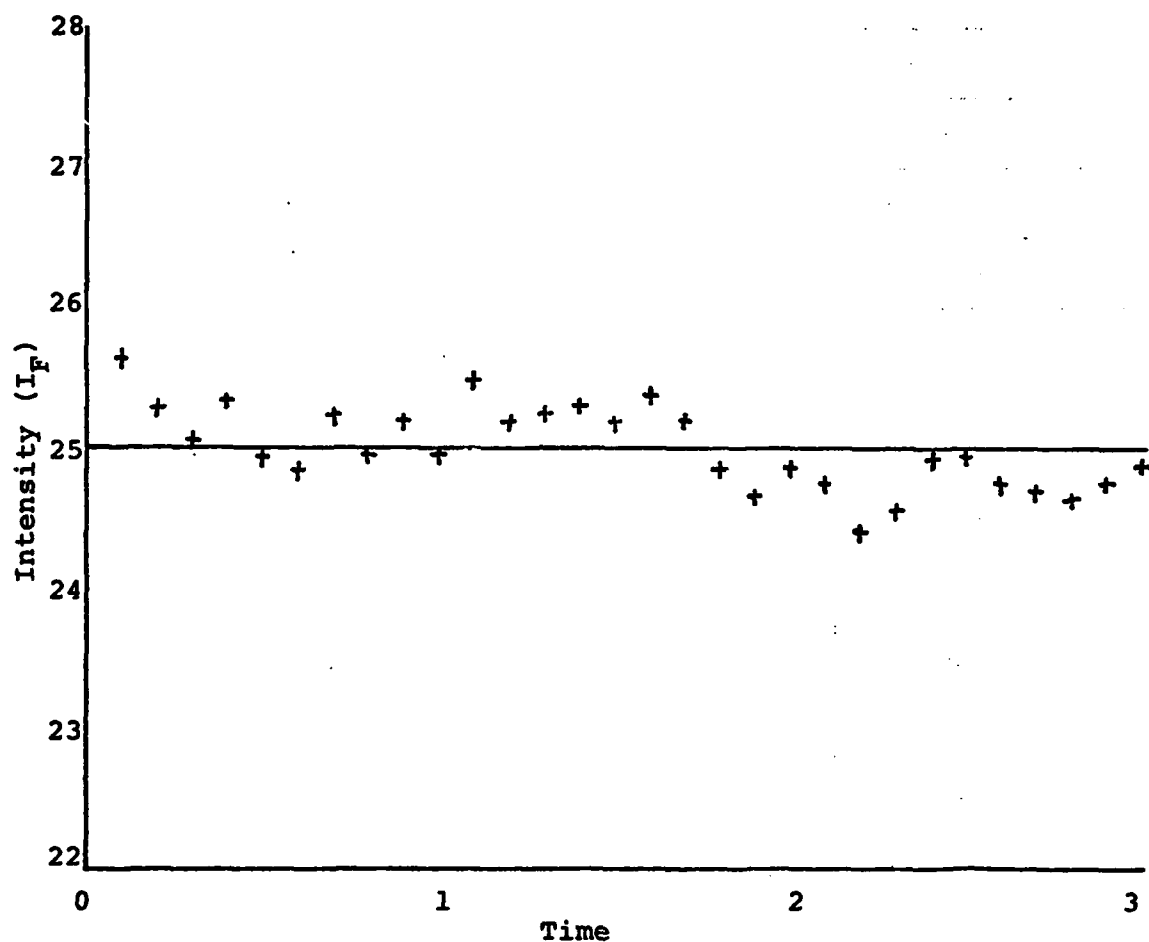


Figure 24. I_{\max} Estimate: Averaged Measurement With Error Function

AR and σ_v Estimation

The gradient minimization of the approximated likelihood function given in equation (131), provided accurate estimation of σ_v and AR. The step size constant c introduced in equations (132) and (133) was determined empirically and set equal to .001. Optimization of this parameter was not accomplished, but .001 yields good response and damping characteristics. The test values of c were between .015 and .0003 inclusive. The larger value provided oscillatory characteristics (underdamped); the estimate would oscillate above and below the average value (about .75 pixels p-p). The smaller value gave more consistent estimates but was slower responding (overdamped). The value used, .001, gave the slightly underdamped response desired.

Figures 25 and 26 are graphs of AR, σ_v , σ_{pv} estimates (σ_{pv} is implied by eq. (119)) for a missile on trajectory 2 (cross range). The truth model values are: $AR_T = 5$, $\sigma_{vT} = 5$, $\sigma_{pvT} = 1$. The response from the filter initial values ($AR = 1$, $\sigma_{vF} = 3$) is rapid and produces values adequate for tracking within .1 seconds. Steady state (when initial transients are gone) is reached in less than .2 seconds and the parameters remain virtually unchanged for the duration of the run.

Note the steady state time averaged values of σ_{vF} and AR_F differ slightly from the truth model values. AR_F is biased downward .93 pixel with a standard deviation of .16 and σ_{vF} is also biased downward .17 with a standard deviation of .32. The standard deviations indicate that the filter values are most likely not equal to the truth model values. The cause for this bias is probably threefold. First is the different way by which the filter evaluates the $h(\underline{x}(t_i), t_i)$ function (equation (123)) and the truth model evaluates the measurement intensity function (equation (7)). Averaging 16 equally spaced points inside the pixel (truth

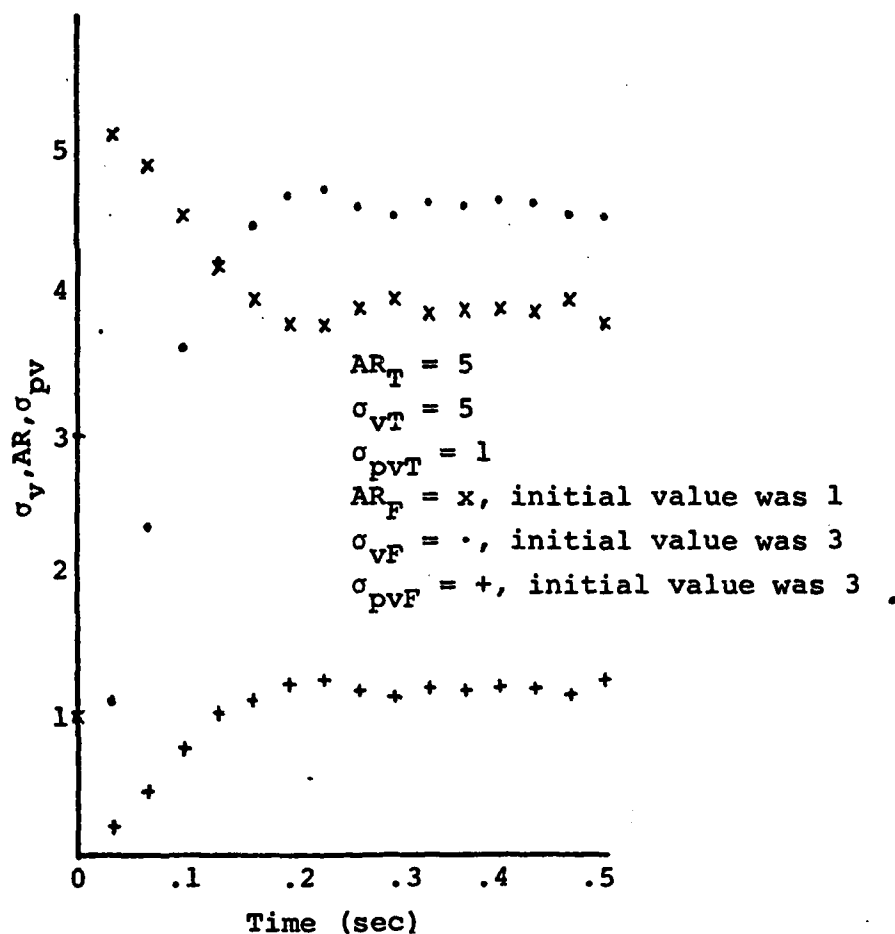


Figure 25. Transient Response of σ_v and AR Estimation

model) yields a much better approximation to the integral than using just the midpoint (filter). Second is that unbiased errors are not assured in this form of the maximum likelihood estimation concept because several simplifications and linearizations are made to the complete maximum likelihood equations to allow real time application. Third, after the truth model image is formed by equation (32), the lowest value in the array (most surely a negative number due to addition of samples of zero mean noise to the pixels at the edge of the array) is set equal to .1. This will

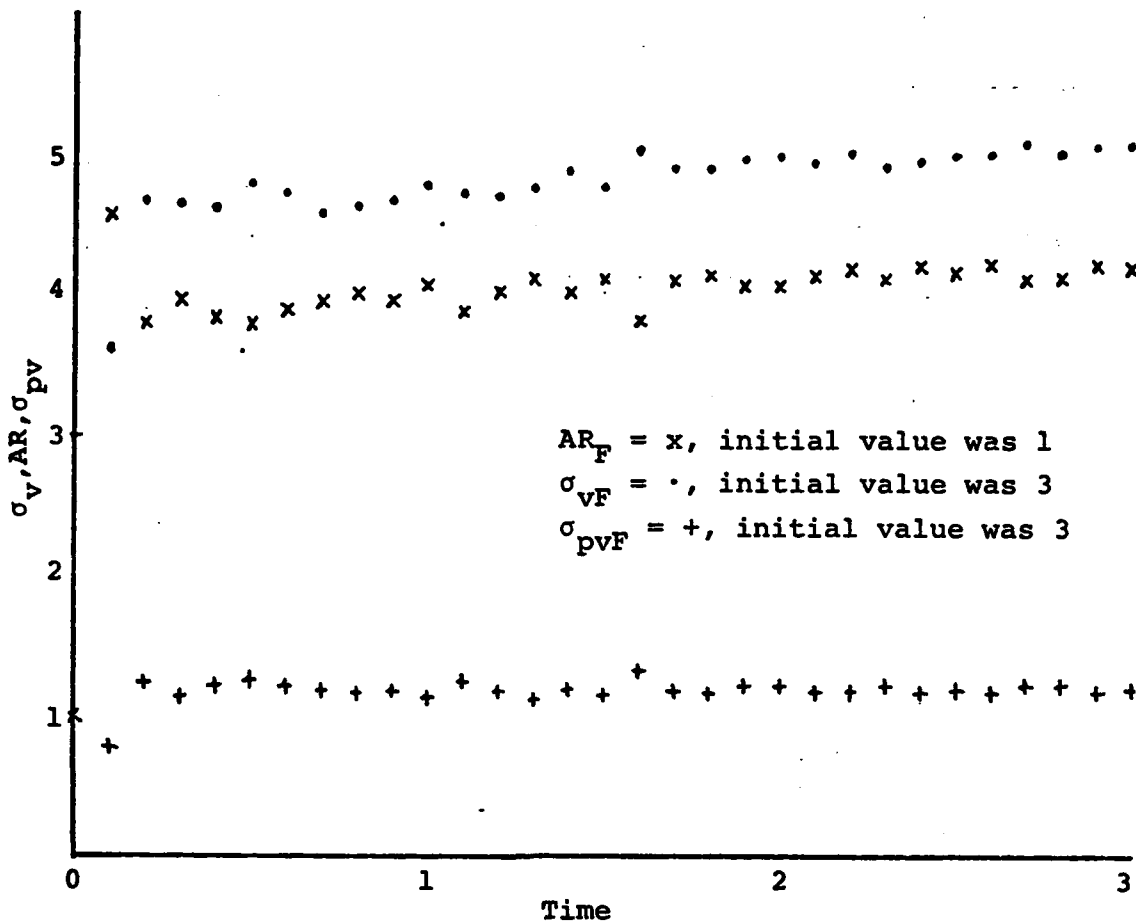


Figure 26. Steady State Response of σ_v and AR Estimation

increase the intensity of each pixel. Since no compensation is added to the filter's estimation of this intensity, any perturbations to the image caused by the truth model simulation procedure will be reflected in the σ_{vF} and AR estimations of the filter.

The original purpose of the σ_v and AR estimation was to enhance state estimation by closely fitting the intensity pattern. As long as this is accomplished, the exact values estimated for σ_{vF} and AR_F are of secondary importance. To this end, an in-depth study to determine the cause of the difference was not accomplished. Figure 27 compares one

[2.25]	[2.71]	[1.78]	[2.11]	[1.54]	[.10]	[1.38]	[2.32]
[.38]	[.43]	[.46]	[.48]	[.48]	[.46]	[.43]	[.38]
[3.42]	[1.89]	[1.47]	[3.80]	[4.07]	[4.57]	[3.46]	[2.05]
[2.70]	[3.03]	[3.27]	[3.39]	[3.39]	[3.27]	[3.04]	[2.71]
[9.82]	[10.26]	[8.74]	[10.91]	[11.63]	[9.28]	[9.29]	[8.25]
[9.87]	[11.06]	[11.93]	[12.40]	[12.41]	[11.95]	[11.09]	[9.91]
[18.49]	[20.42]	[21.86]	[22.73]	[22.83]	[23.01]	[21.17]	[21.40]
[18.66]	[20.91]	[22.56]	[23.44]	[23.45]	[22.60]	[20.97]	[18.74]
[18.21]	[20.63]	[22.13]	[23.16]	[23.22]	[20.70]	[21.54]	[18.28]
[18.25]	[20.45]	[22.07]	[22.93]	[22.95]	[22.11]	[20.52]	[18.33]
[9.51]	[8.22]	[11.07]	[9.83]	[10.22]	[10.19]	[7.59]	[7.04]
[9.24]	[10.35]	[11.17]	[11.61]	[11.62]	[11.19]	[10.39]	[9.28]
[2.61]	[2.73]	[2.20]	[1.98]	[4.65]	[3.76]	[2.27]	[1.98]
[2.42]	[2.71]	[2.93]	[3.04]	[3.04]	[2.93]	[2.72]	[2.43]
[1.80]	[2.06]	[.24]	[1.23]	[1.75]	[.24]	[3.17]	[.47]
[.33]	[.37]	[.40]	[.41]	[.41]	[.40]	[.37]	[.33]

Truth model parameters: $\sigma_v=5.0$, $\sigma_{pv}=1.0$, $AR=5.0$

Filter estimates: $\sigma_v=5.15$, $\sigma_{pv}=1.23$, $AR=4.18$

The image intensity is the upper number of the pair and the filter estimate is the lower number.

Figure 27. Truth Model and Filter Image Comparison

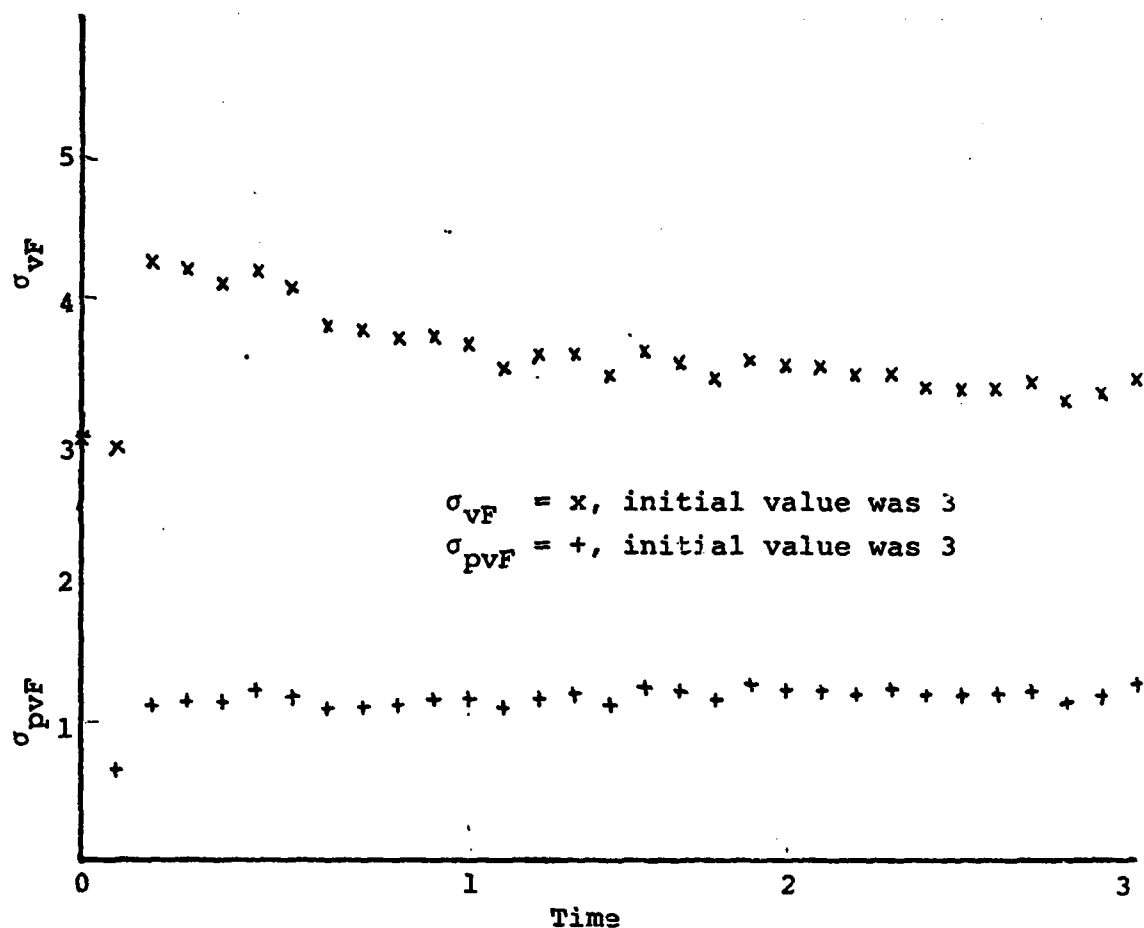


Figure 28. σ_v and σ_{pv} for $AR = 3$ (Truth Model)

typical frame of the simulated image to the filter's estimate of that image. The close fit between the two, especially near the intensity peak, is evident. Truth model parameters for this frame were $\sigma_{vT} = 5.0$, $\sigma_{pvT} = 1$, $AR_T = 5$, $I_{\max} = 25$ and filter estimated parameters were $\sigma_{vF} = 5.15$, $\sigma_{pvF} = 1.23$, $AR_F = 4.18$ and $I_F = 25.33$. Also, the truth model image is corrupted with noise as usual.

Figure 28 is a graph of σ_{vF} and σ_{pvF} estimates with truth model $AR_T = 3$, $\sigma_{pvT} = 1$, $\sigma_{vT} = 3$. Figure 29 is similar to Figure 28 with truth model $AR = 1$, $\sigma_{pv} = 1$, $\sigma_v = 1$. In the latter case of Figure 29, the σ_{vF} and σ_{pvF} are erratic due to a loss of resolution when σ_v and σ_{pv} are less than

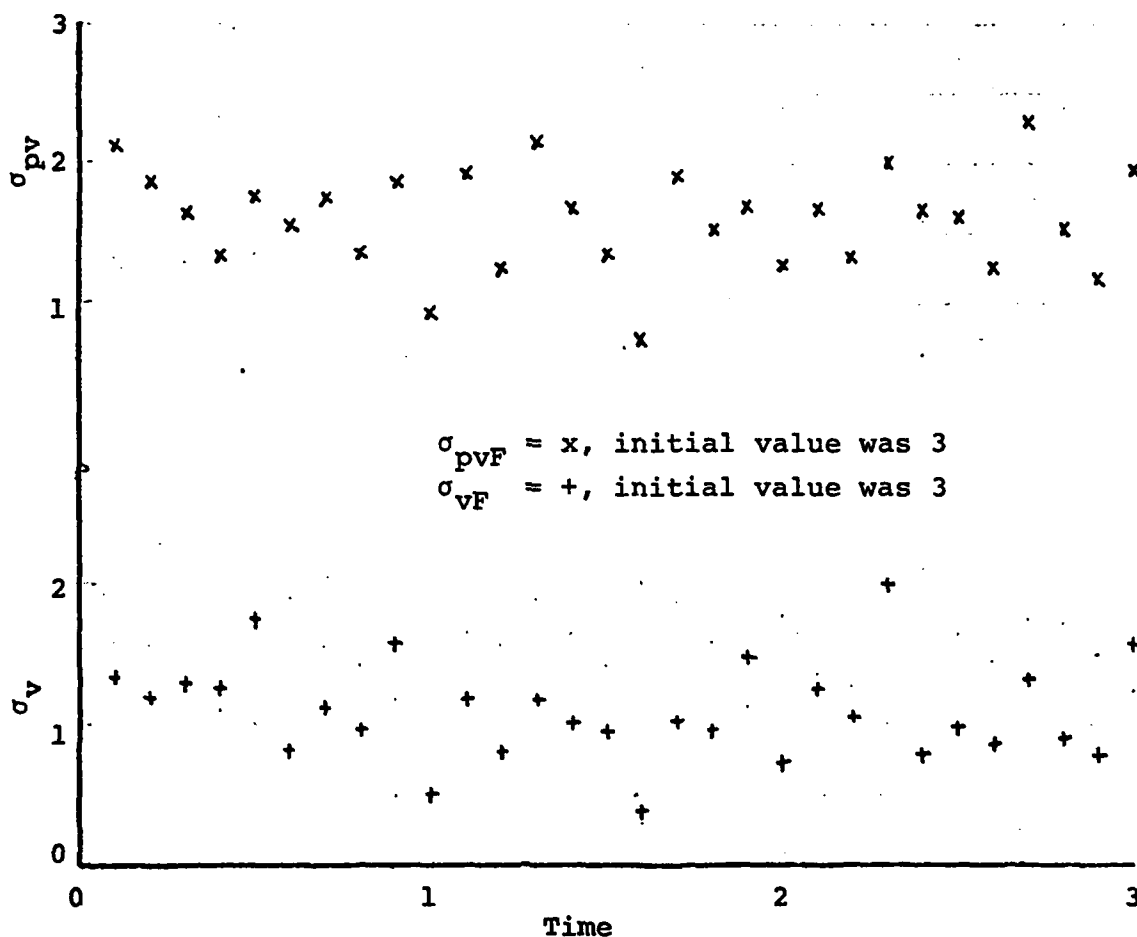


Figure 29. σ_v and σ_{pv} for AR = 1 (Truth Model)

two. The area of the array where the larger gradients exist simply are not covered by enough pixels to provide smooth estimates. This erratic behavior, although uneasy, did not degrade filter performance noticeably. Time averaging could be used to smooth the estimates if required by different circumstances.

Target Acquisition Procedures

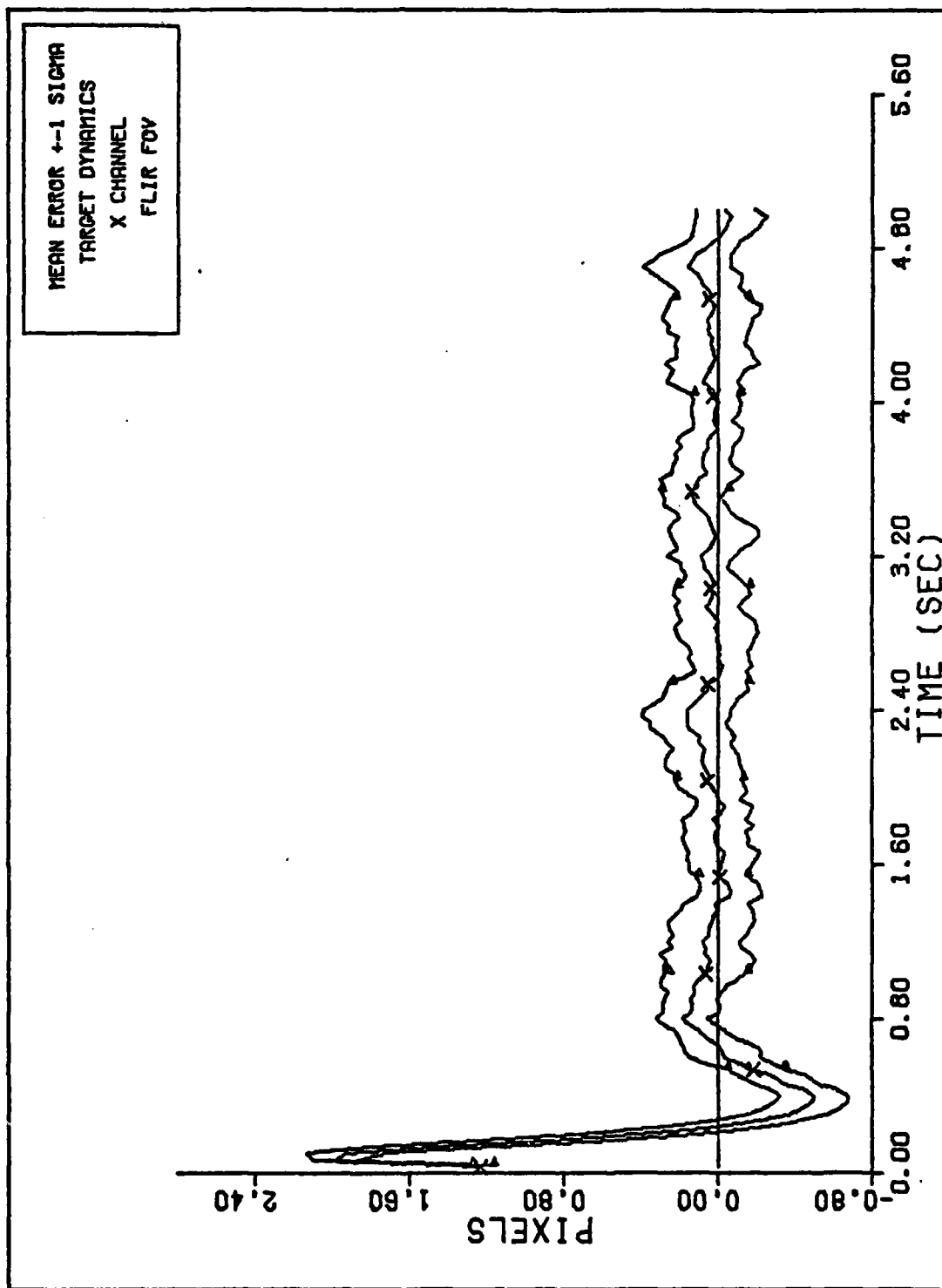
The acquisition procedures as described in Chapter IV worked well under nominal conditions and received limited testing to determine their flexibility. The acquisition

strategy basically focuses on two questions: to what value should \underline{P}_O be set, and how should \underline{Q}_{Fd} be used for acquisition?

For most cases, the acquisition procedure was to set both \underline{P}_O and \underline{Q}_{Fd} large and then to use a linearly decreasing \underline{Q}_{Fd} . The values of \underline{P}_O were large to reflect the inaccuracies of the handoff values from other tracking systems. The variances for the position, velocity, and accelerations (when eight state filter was used) on the main diagonal were set to 25 (pixels²), 2000 (pixels²/sec²), and 100 (pixels²/sec⁴), respectively. The variance of the atmospheric jitter was kept at its initial input value of .2 (pixels²). Except for the main diagonal, \underline{P}_O contained only zero entries. In simple terms, the high values in \underline{P}_O cause the filter initially to follow the measurements more closely through an increase in the gain matrix. Also for acquisition, σ_D^2 , used in computing \underline{Q}_{Fd} for the eight state filter per equation (107) and for the six state filter per equation (118), is set initially large at 600 (pixels/sec³ for the six state filter, pixels/sec⁵ for the eight state filter). This value was chosen based on observed values in six state filter simulations. This value was arbitrarily used with the eight state filter and retained because it gave acceptable performance. The value of σ_D^2 (the target dynamics portion of \underline{Q}_{Fd}) was decreased linearly until $\sigma_D^2 \cong 150$ (pixels/sec³ or pixels/sec⁵ depending on the filter used) between times 0.0 and 0.5 seconds. The high initial values in \underline{Q}_{Fd} reflect an uncertainty in the target dynamics upon acquisition. At hand-off the missile could be turning, linearly accelerating, or in constant velocity straight flight, and the tracker must be prepared to handle any case. Because \underline{Q}_{Fd} effects \underline{P} through equation (108) and consequently the gain, the initially large \underline{Q}_{Fd} again causes the filter to strongly track the measurements though less directly than large \underline{P}_O . This can be viewed as extending

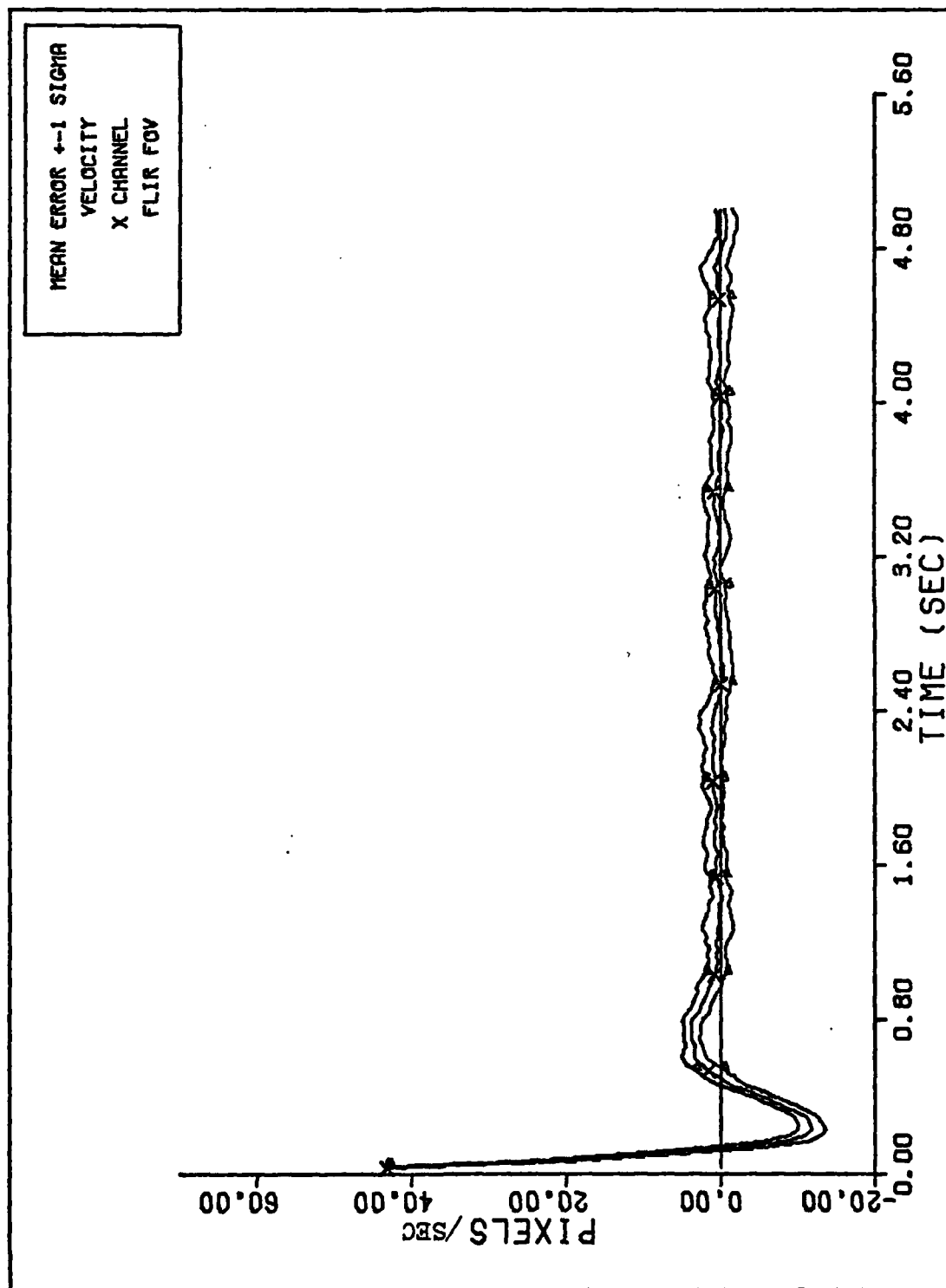
the acquisition (maintaining high \underline{P} and \underline{K}) time interval beyond that obtained when a high \underline{P}_0 is used with a low constant \underline{Q}_{Fd} for all time. The linear slope used for \underline{Q}_{Fd} was based on intention; better schedules are probably devisable with more extensive study. Overall, these two simple techniques performed well in cases 33 through 35 in which these techniques were used in conjunction with the six state filter, as can be viewed in their plots in Appendix L. Because these cases were not directly designed to test the acquisition process (i.e. hand-off values of velocities and position to the filter had no errors), little can be concluded from them other than the fact that the acquisition schemes worked.

One test of the filter's acquisition flexibility, case 32, was performed and demonstrated good filter stability. The test scenario was the tracking of a target moving cross range using the eight state filter. The range at closest approach was 10 kilometers. The high \underline{P}_0 scheme previously discussed was used, but the linear schedule for \underline{Q}_{Fd} was not used because it was linked in the computer software to the \underline{Q}_{Fd} estimation scheme which was deactivated for this case. Rather, \underline{Q}_{Fd} was held constant throughout the runs with $\sigma_D^2 = 600$ (pixels/sec⁵). The hand-off values of the inertial velocity were $\dot{x}_0 = -508$ meters/sec, $\dot{y}_0 = -308$ meters/sec, and $\dot{z}_0 = 8$ meters/sec compared to true values of -500, -300 and 0 meters/sec. These values were chosen to be representative of probable hand-off errors from instruments such as radars and constitute a 2.4 percent error. The results of case 32 show the flexibility of these procedures, as can be seen in Figures 30 and 31. Figure 30 shows the x channel target position mean error and one sigma envelope. The plot shows an underdamped but stable response to the initial velocity errors. Similarly, Figure 31 displays the x channel, target velocity, mean error and 1 sigma



X CHANNEL DYNAMICS ERROR (S/N=12.5)

Figure 30. Case 32: Target Position Mean Error and 1 Sigma Envelope



X CHANNEL VELOCITY ERROR (S/N=12.5)

Figure 31. Case 32: Target Velocity Mean Error and 1 Sigma Envelope

envelope. The filters response is again stable though underdamped. Similarly, results were achieved for y channel target position and velocity errors (see Appendix L). The target was never lost in the 20 simulations that constitute case 32. These results give some confidence in the high P_0 acquisition technique and indicate that the linearly decreasing Q_{Fd} scheme is not needed for successful target acquisition. However, much more study is needed to determine the limitations of this acquisition technique or to develop other techniques. Further study is desirable but infeasible in the time allotted. The basic feasibility and performance of these techniques have been demonstrated. Another case which warrants attention in resolving the acquisition strategy would rerun case 32 (hand-off errors) without increased P_0 to determine how critical it is to successful acquisition.

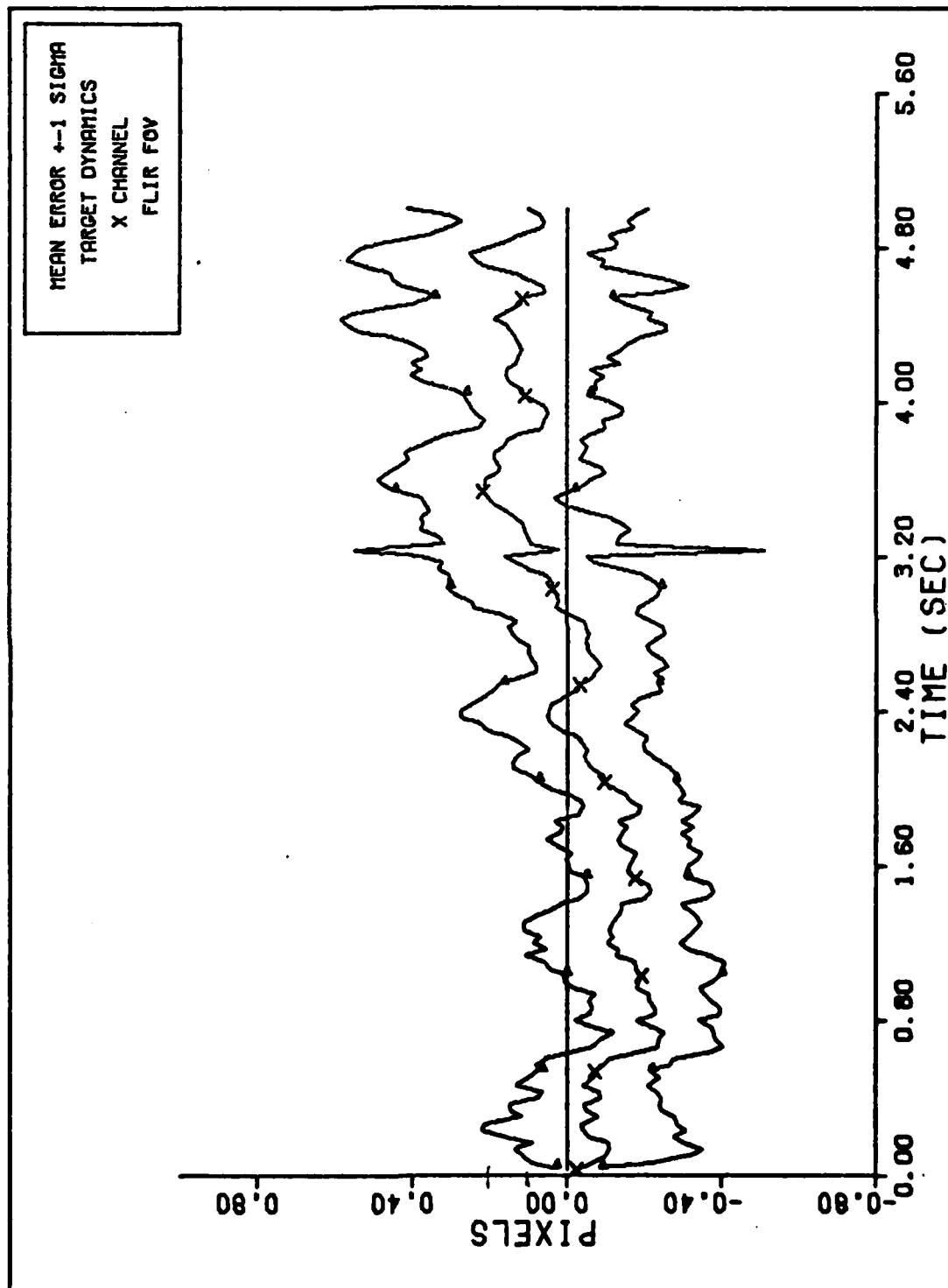
Performance of Target Dynamics Equations

The two formulations of target dynamics which resulted in six state and eight state filters were tested to establish their performance capabilities. The six state filter was first tested against the missile moving cross range (see Chapter II for motion descriptions), and the resulting performance plots indicated a serious problem attributable to a non-inertial acceleration effect caused by the rotating reference frame. A circular flight path for the target was tested against the six state filter to confirm that the problem was indeed caused by non-inertial acceleration effects of the rotating reference frame, as will be discussed in more detail in the subsequent paragraphs of this section. The eight state filter, as expected, solved this problem by estimating the acceleration, which compensates for the rotating frame effects.

Case 33 shows the six state filter tracking the target moving diagonally cross range starting towards the lower

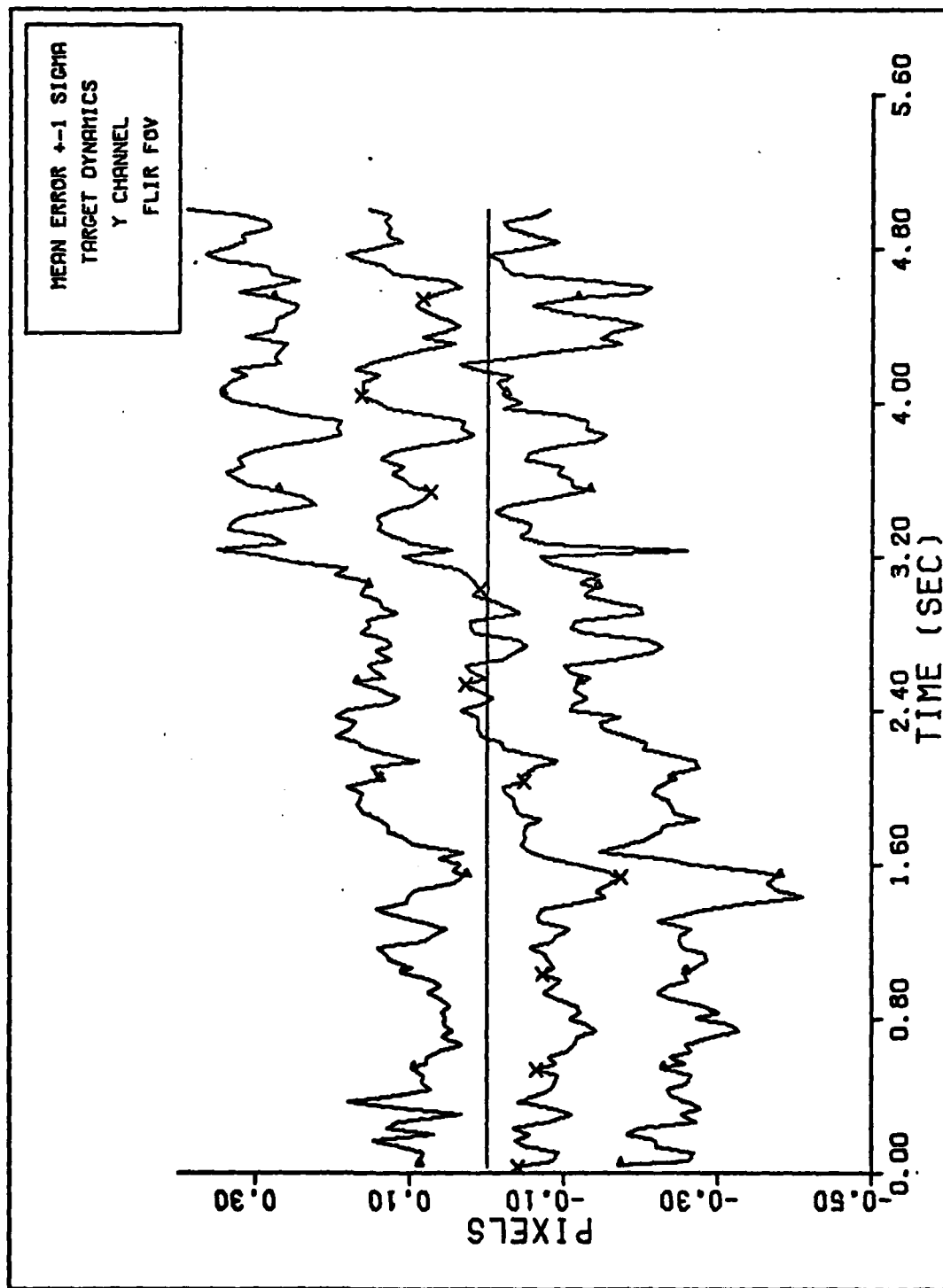
left hand corner of each frame. \hat{Q}_{Fd} estimation is active during this case without ad hoc \hat{Q}_{Fd} bounding (as will be explained more fully in the next section). Figures 32 and 33 show the x and y channel target position error performance plot for case 33. The projection into the FLIR image plane of the inertially constant velocity vector of the missile changes with time as the FLIR system and its reference frame rotates to maintain the target in the center of the field of view. Thus, a non-inertial acceleration is created by the rotating system and frame. This unmodeled non-inertial acceleration manifests itself in a small but definite diagonal trend (positive slope) in the mean, and mean plus and minus one sigma curves of Figures 32 and 33. The mean value curve is no longer centered about the 0 line but about a line with positive slope (about 5 to 10 degree slope) which intersects the 0 line about 2.5 seconds, the point of minimum range. Figures 34 and 35 show the x and y channel target velocity error performance plots for case 33. The positive diagonal trend is again noted, and the velocity estimate diverges during the last second of the run because the \hat{Q}_{Fd} estimation does not keep up when the non-inertial acceleration again becomes large. The acceleration created by the rotating reference frame is equal to two times range rate times angular velocity divided by range, $2\dot{\rho}\omega/\rho$ where ω is the angular velocity of the LOS. This expression is the angular equivalent to the linear translational term found in an equation in Reference 15, page 31. Thus, the non-inertial acceleration becomes insignificant at very long range, and at minimum range for the straight cross range target because $\dot{r} = 0$. (Note that ω is at its maximum at minimum range.) The maximum absolute acceleration occurs somewhere between these two positions and is a function of the three quantities, ρ , $\dot{\rho}$, and ω .

To confirm the hypothesis that the unmodeled non-inertial acceleration was causing the trends shown in case 33, cases 34



X CHANNEL DYNAMICS ERROR (S/N=12.5)

Figure 32. Case 33: Target Position Mean Error and 1 Sigma Envelope



Y CHANNEL DYNAMICS ERROR (S/N=12.5)
Figure 33. Case 33: Target Position Mean Error and
1 Sigma Envelope

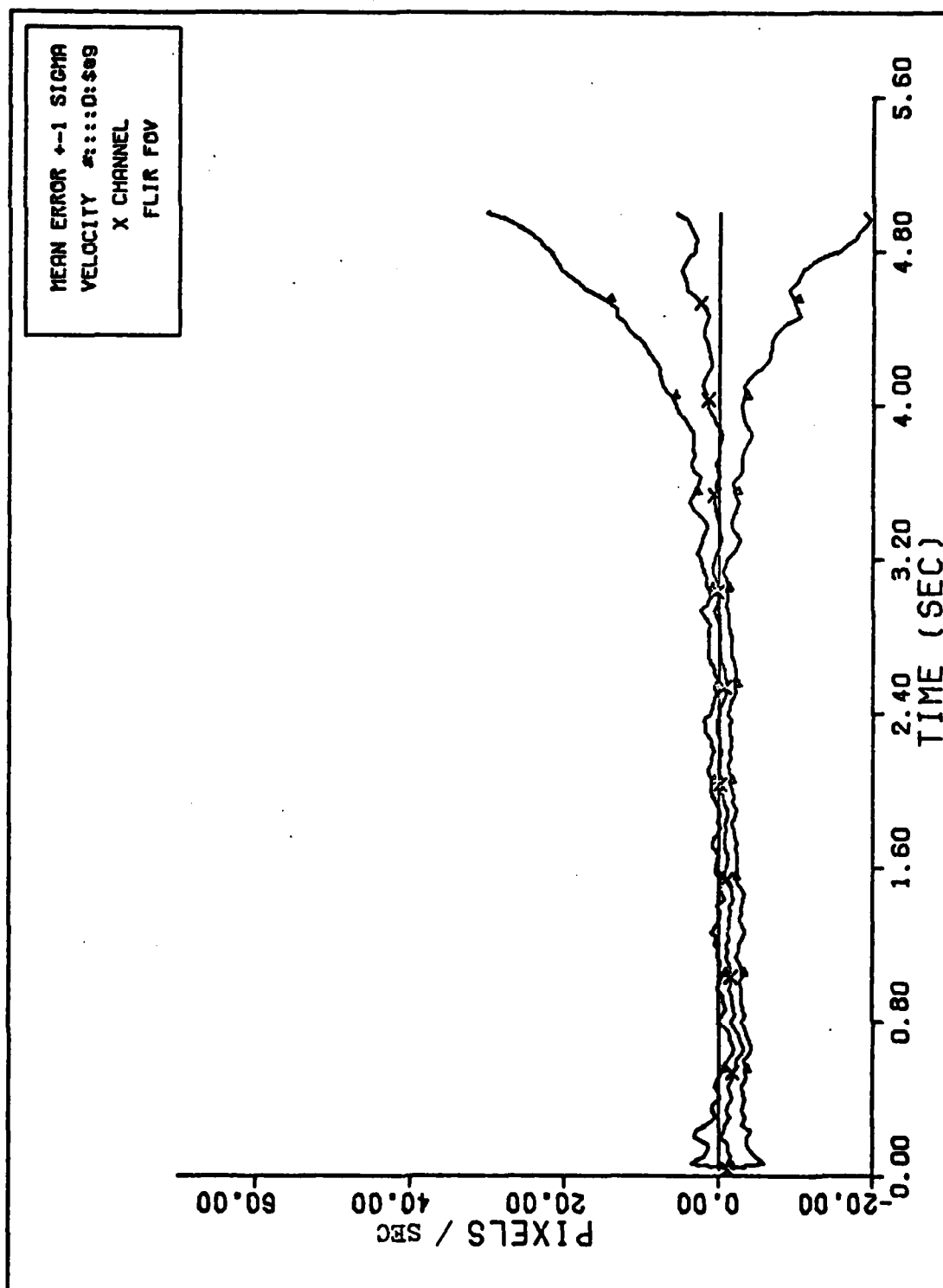


Figure 34. Case 33: Target Velocity Mean Error and 1 Sigma Envelope

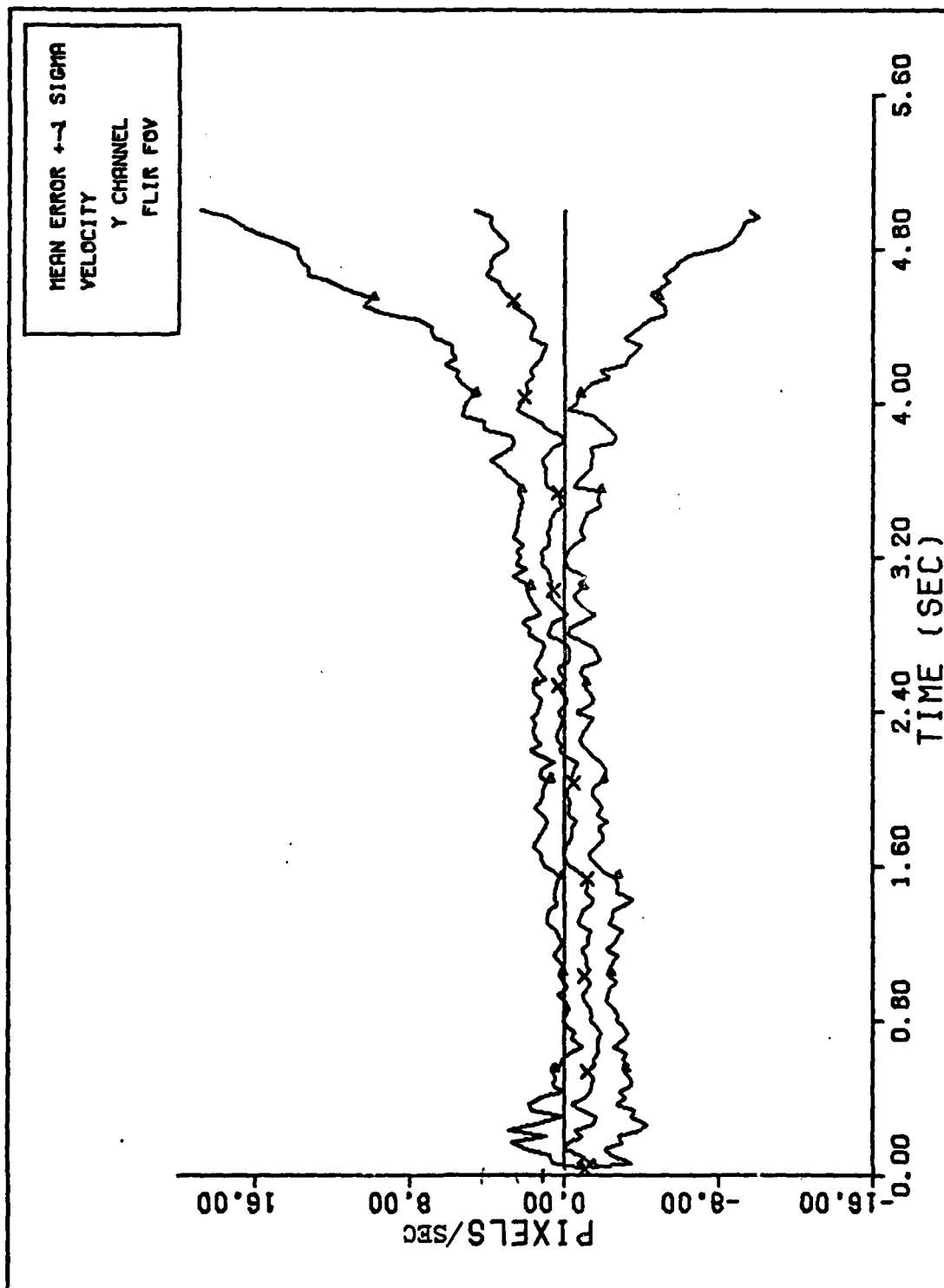


Figure 35. Case 33: Target Velocity Mean Error and 1 Sigma Envelope - 4 Channel)

and 35 were run. The target flight path as viewed from above followed a large circle with a radius of 20 kilometers and with the tracking system located at the circle's center. Such a flight path would not result in any non-inertial accelerations because the true target velocity vector would have a constant projection in the plane of the FLIR field of view rotating with the target. The six state filter tracks the target extremely well. However, the appearance of negative values on the main diagonal of \underline{P} due to numerics and the \underline{Q}_{Fd} estimation caused some problems, and several ad hoc fixes were tried as will be discussed in more detail in the next section.

Thus, cases 34 and 35 represent the tracking of the circular flight path target with different solutions to the \underline{P} problem. In case 34 the main diagonal terms of the newly computed \underline{Q}_{Fdl} (the second term on the right hand side of equation (144)) were set to .1 when a negative value was detected. The other values in the row and column in which the negative main diagonal term resided were set to zero. Similarly, for case 35 any negative values on the main diagonal of \underline{P} were set to .1 and the corresponding rows and columns were zeroed out. Figures 36 and 37 show the x and y coordinate target position error statistics for case 34. Gone is the diagonal trend in the statistics which had been noted in Figures 32 and 33 for case 33. Figures 38 and 39 display the x and y channel target velocity errors for case 34, and the non-inertial acceleration effects are absent, too. Case 35 results exhibited the same characteristics, with the disappearance of the trends seen previously in case 33.

The eight state filter, which includes acceleration states for the x and y channels, was a logical solution to the problem of non-inertial accelerations of the rotating reference frame. An altered dynamics model of the six state filter incorporating appropriate correction terms is a more direct approach to deal with the problem, but it would

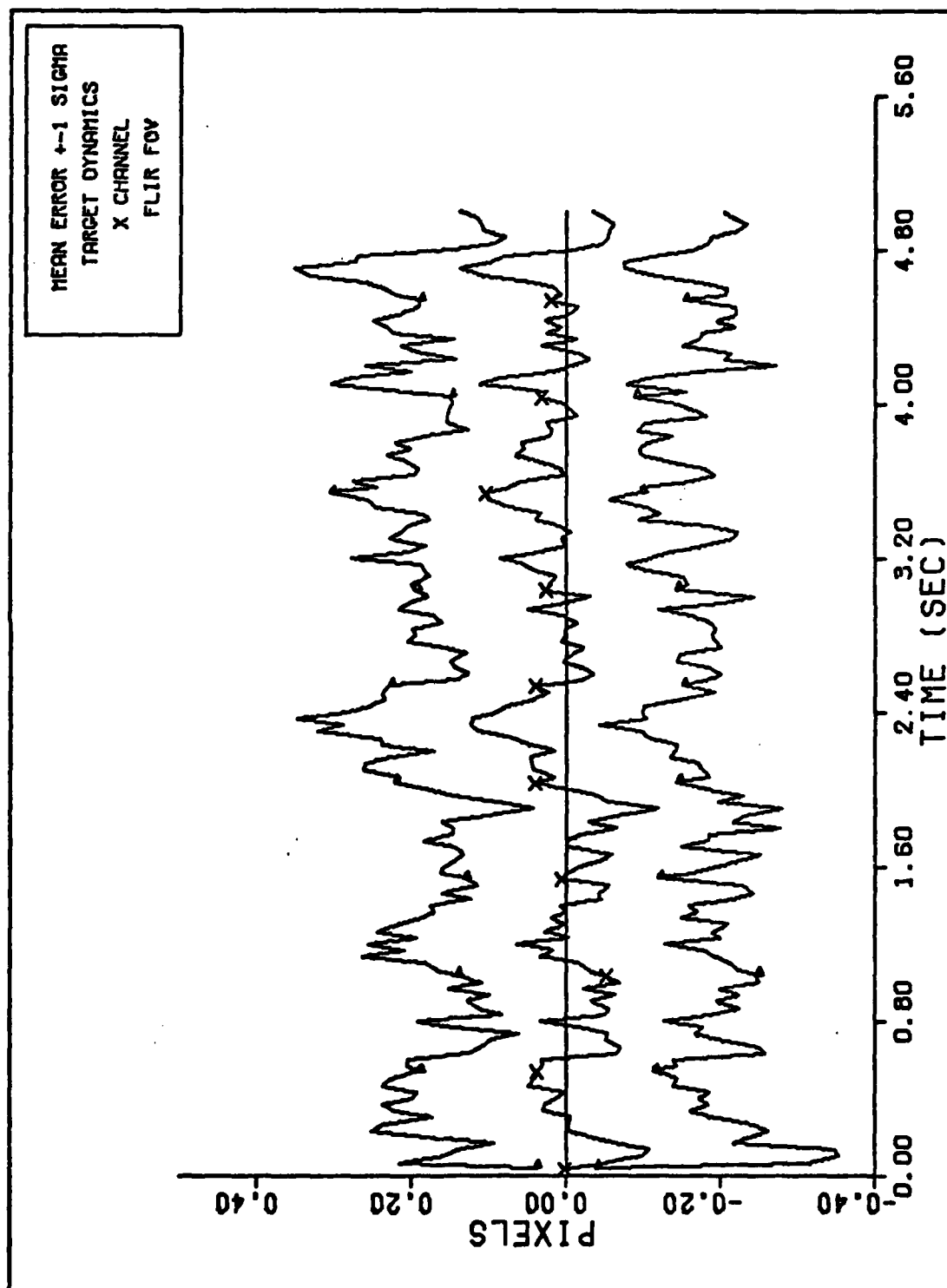
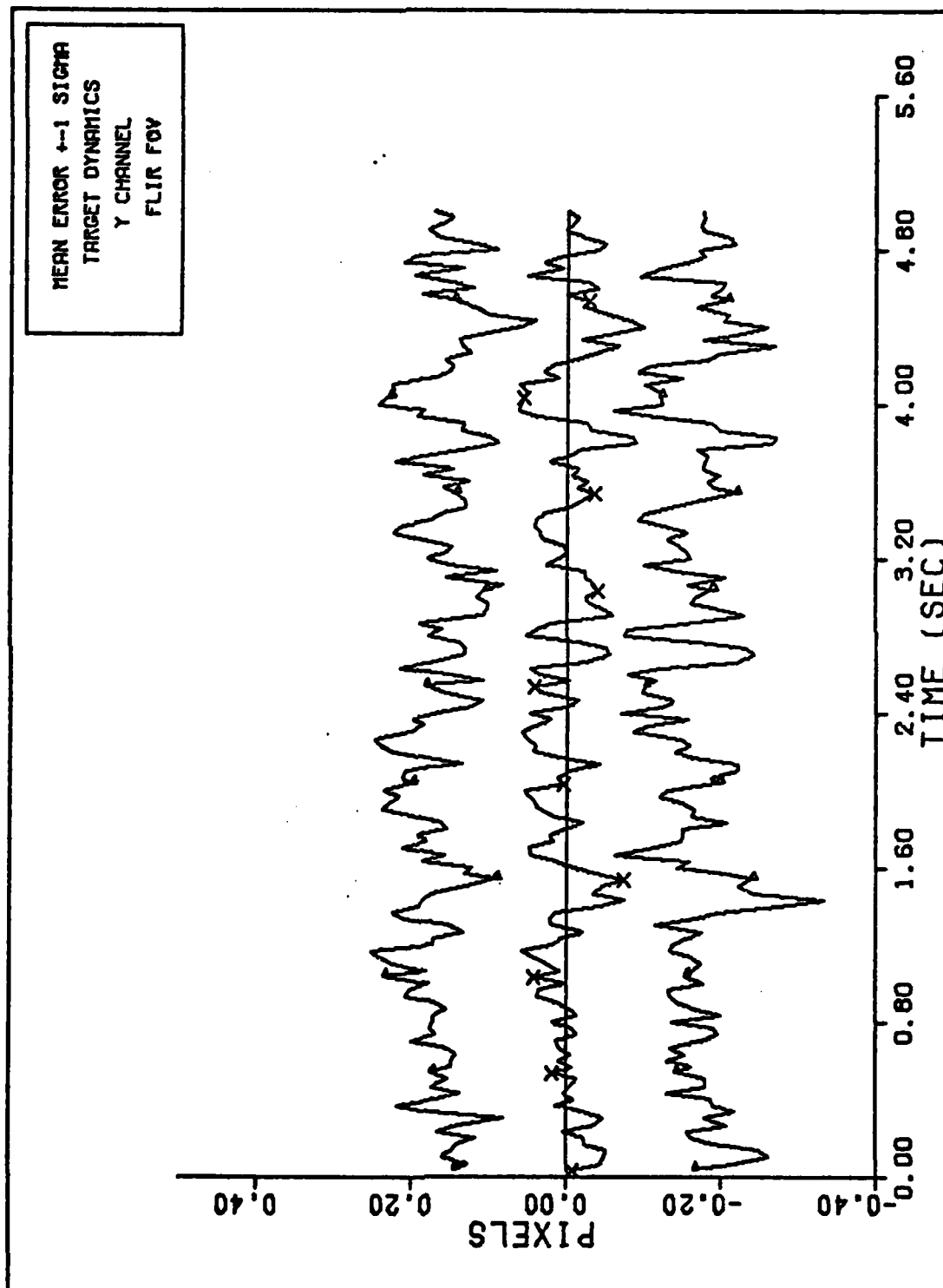


Figure 36. Case 34: Target Position Mean Error and 1 Sigma Envelope



Y CHANNEL DYNAMICS ERROR (S/N=12.5)
Figure 37. Case 34: Target Position Mean Error and
1 Sigma Envelope

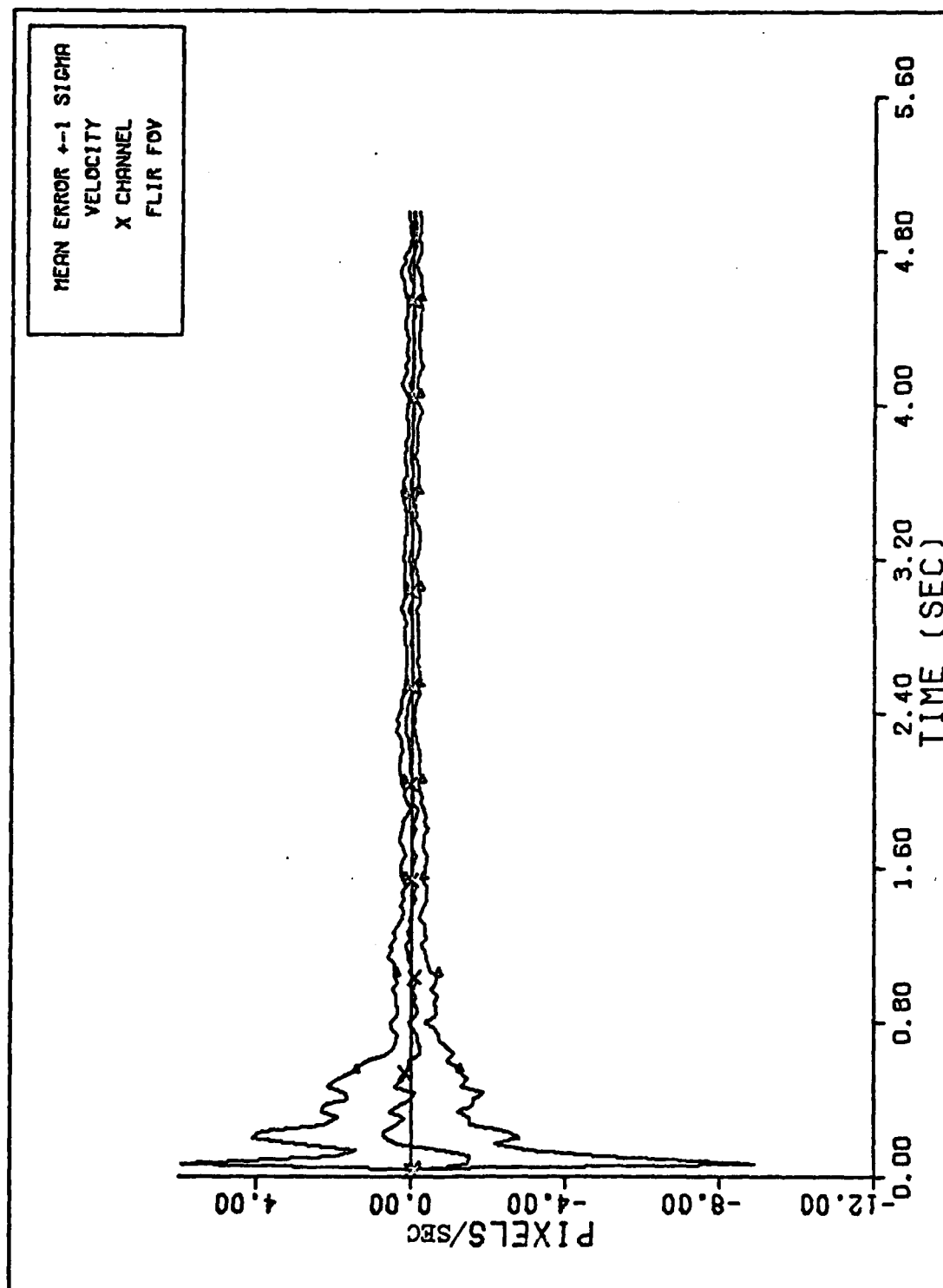


Figure 38. Case 34: Target Velocity Mean Error and
1 Sigma Envelope

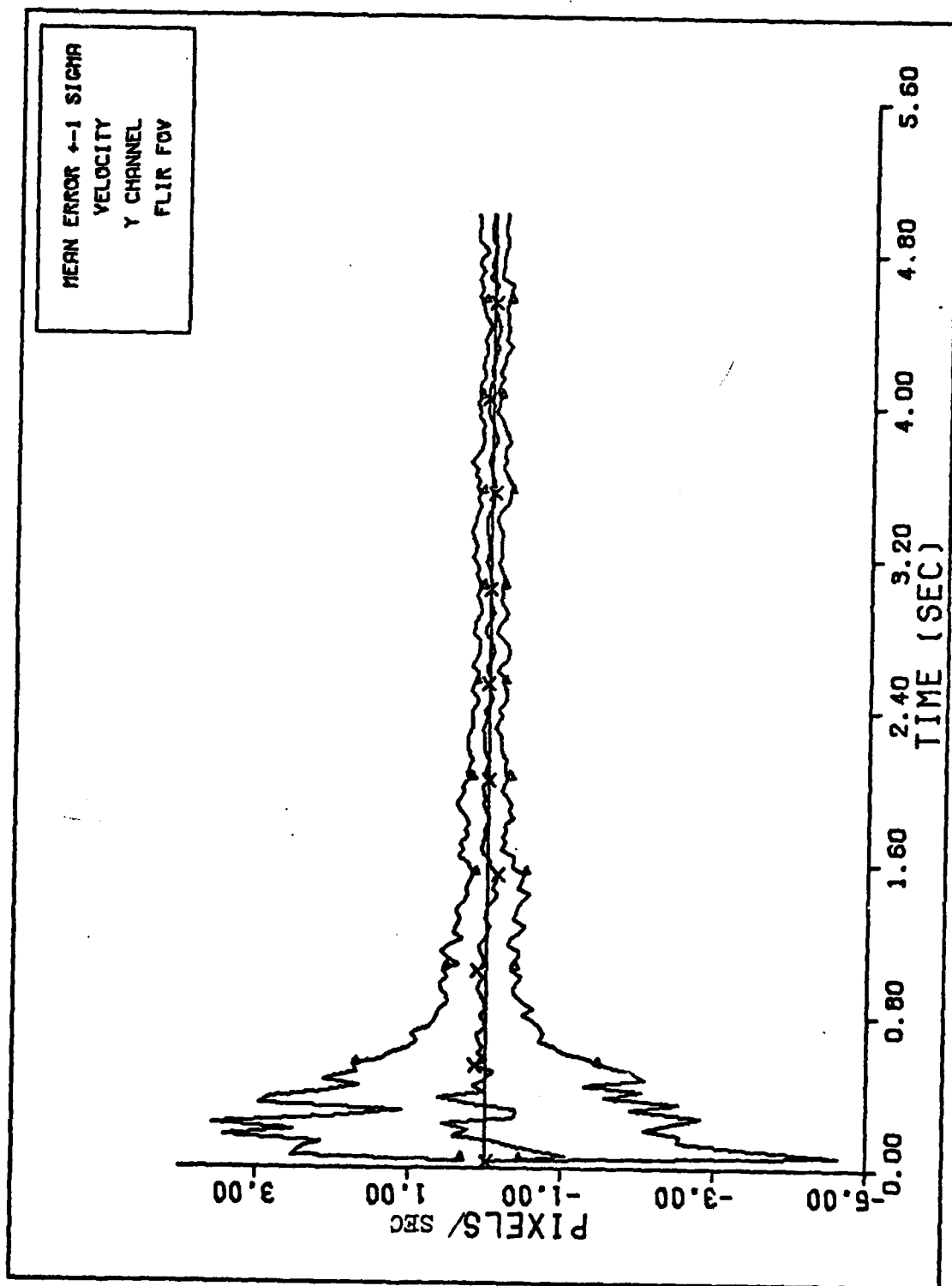
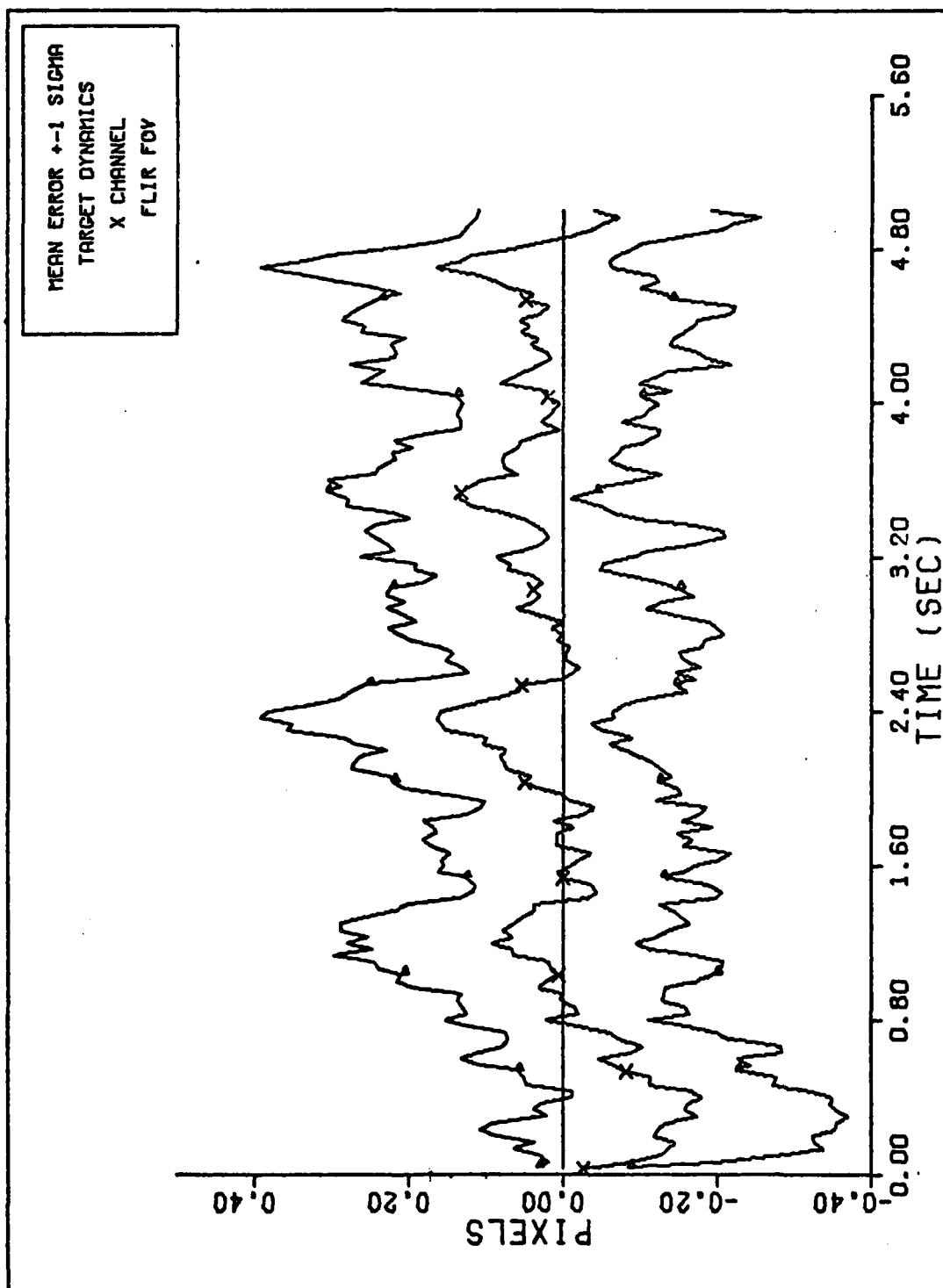


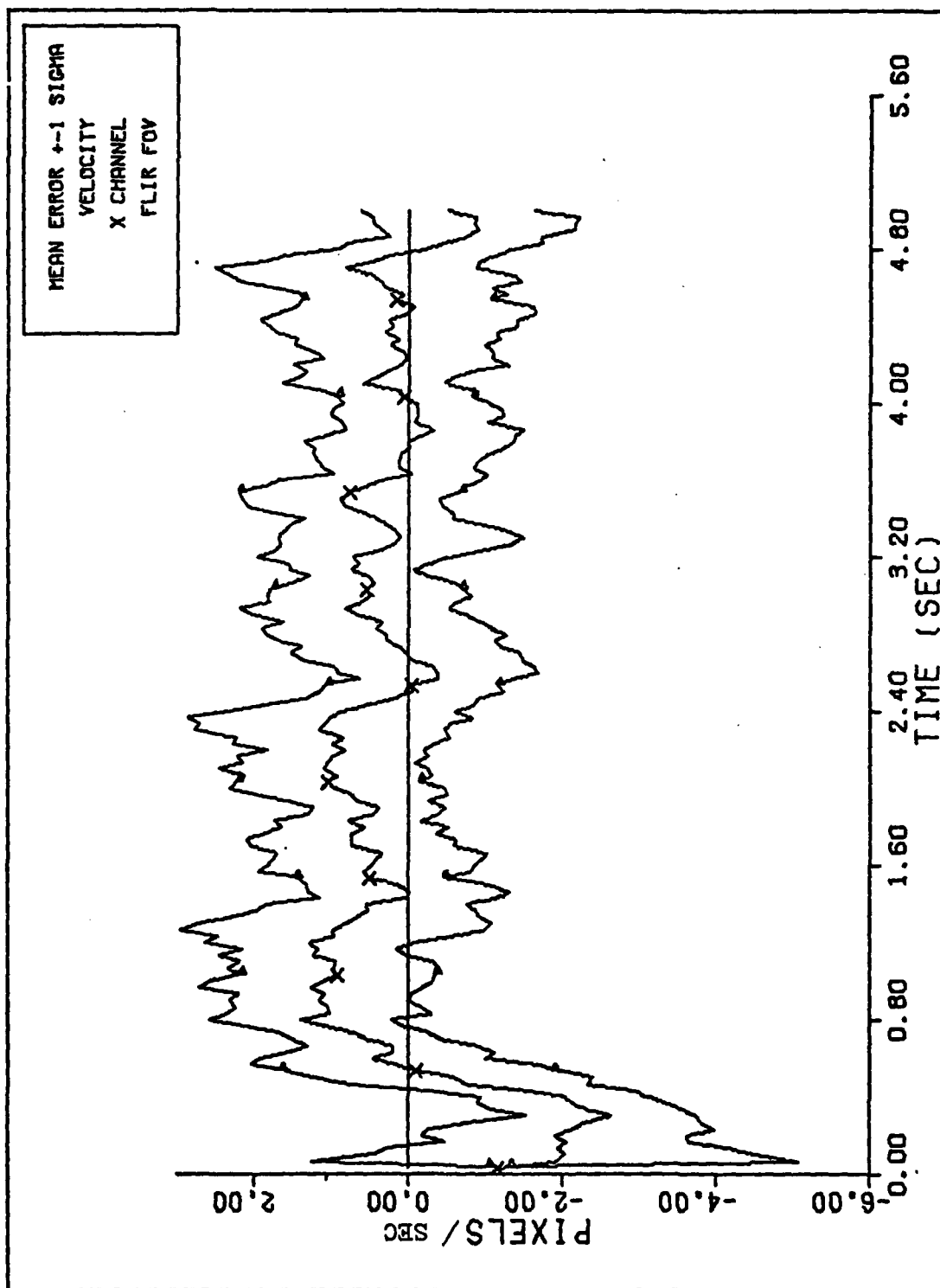
Figure 39. Case 34: Target Velocity Mean Error and 1 Sigma Envelope - 4 Channel)

require range and range rate data to work effectively and would result in a nonlinear propagation formulation which would be more difficult to use than the eight state filter. In case 36, the eight state filter tracks a target following the cross range trajectory (the same trajectory as cases 32 and 33). \underline{Q}_{Fd} was held constant throughout the simulations -- a \underline{Q}_{Fd} acquisition schedule was not used and \underline{Q}_{Fd} adaptive estimation not incorporated.

Figures 40 and 41 show the x channel target position and velocity errors, respectively. The non-inertial acceleration effects due to the rotating reference frame are now estimated in the filter so that the diagonal trend of the statistics seen in Figures 27 through 30 of case 33 is not seen here. For a more direct comparison to case 36, cases 33, 34, and 36 should be rerun without \underline{Q}_{Fd} adaptation.



X CHANNEL DYNAMICS ERROR (S/N=12.5)
Figure 40. Case 36: Target Position Mean Error and
1 Sigma Envelope



X CHANNEL VELOCITY ERROR (S/N=12.5)
Figure 41. Case 36: Target Velocity Mean Error and
1 Sigma Envelope

Q_{Fd} Estimation Performance Analysis

The estimation of Q_{Fd} as formulated in Chapter IV met very limited success for the trajectories tested and eventually became more hindrance than help in terms of reaching the research goals. Q_{Fd} estimation was intended to allow filter adaptation to changes in target dynamics. However, it did not prove to be responsive enough to prevent loss of track in high g turns. Other solutions to the high g turns were investigated as will be discussed in the next section. Filter stability problems also were caused by the Q_{Fd} estimation technique for certain target scenarios. The first cases of interest are 33 and 37 in which a target moving straight cross range is tracked by the six state filter with and without Q_{Fd} estimation, respectively. A summary of the filter performance for both cases is found in Table XVI. The estimation scheme worked successfully in case 33, resulting in very acceptable results slightly larger than those achieved without Q_{Fd} estimation. Because Q_{Fd} estimation is aimed at adaptation to changing missile dynamics, it was not expected to improve on steady state (non-accelerating missile) filter performance unless there was a significant mismatch in non-adaptive Q_{Fd} parameters. The non-adaptive Q_{Fd} six state filter was manually tuned to this particular cross range trajectory. Consequently, its performance was very good. Again, time constraints did not allow an extensive study of trajectories with increased motion (linear accelerations and a variety of turns) creating significant mismatches for the non-adaptive Q_{Fd} filter.

As mentioned briefly in the previous section, the study of the unmodeled non-inertial acceleration terms, which was noted as a performance degrading trend in case 33, led to the study of the circular flight path trajectory tracked by the six state filter in cases 34 and 35. Then the eight state filter was tested against the straight cross range target in case 36. These cases, 34, 35 and 36, were initially

TABLE XVI

Six State Filter Tracking Cross Range Target
 With and Without Q_{Fd} Estimation -
 Cases 33 and 37

	Case 33		Case 37	
	w/ Q_{Fd} est.		w/o Q_{Fd} est.	
	x	y	x	y
Average Standard Deviation				
Target Position				
Actual	.22	.19	.20	.17
Filter-Indicated	.22	.20	.22	.21
Target Velocity				
Actual	3.5	3.0	2.5	7.5
Filter-Indicated	3.0	3.0	5.0	5.0
Average Absolute Error				
Target Position	.12	.08	.05	.05
Target Velocity	1.5	1.5	1.0	1.0

plagued with the infrequent appearance of negative values on the main diagonal of \underline{P} due to the Q_{Fd} estimation. What seems to happen is that when the dynamics model matches the target trajectory (i.e. six state filter with circular trajectory and eight state filter with straight cross range trajectory), values in \underline{P} become small and susceptible to going negative. This can be caused by errors in using $\Delta x \Delta x^T$ as a representation of \underline{KHP} in equation (138).

Several solutions to the \underline{P} problem were tried before final ad hoc fixes were found. The initial theory was that the main diagonal terms of \underline{Q}_{Fd1} , the second term on the right hand side of equation (144), should be bounded above at specified values and below at zero. The lower bounding was accomplished by testing the main diagonal terms of \underline{Q}_{Fd1} for each sample period. If a negative value was found, that value was set to zero as well as every value in the row and column in which it was found. For example, if the third term along the main diagonal was found to be negative, then

all the values in the third column and all the values in the third row were set to zero. If the computation of Q_{Fd1} resulted in a main diagonal value greater than a prespecified value, say 600, then that value was set to 600 and the other matrix values in the row and column of the main diagonal term were set to appropriate proportionate values (maintaining same correlation coefficients through the adjustment). For example, the third main diagonal term is computed to be 700. Then that term is set to 600 and the values of the third column and row were multiplied by the square root of $600/700$. However, these two techniques did not totally remedy the negative values on the main diagonal of P . The same lower bounding scheme used on Q_{Fd1} was attempted with P . The result was a totally unacceptable tendency for the filter to lose track of the target (go unstable). Finally, two similar ad hoc procedures were tested and appeared to work. One technique performs the same lower bounding process on Q_{Fd1} as described above except that a specified positive value, such as .1, is substituted for the main diagonal bound instead of zero. The other technique is the same as just described but performed on P . Lower bounding Q_{Fd1} and P at .1 was used successfully in cases 34 and 35, respectively, as depicted previously in Figures 31 to 34. The value of .1 was determined to be a reasonable lower bound based on the observed values of P for a variety of cases. Comparable cases using zero lower bounding on Q_{Fd1} and P and upper bounding on Q_{Fd1} were not run because usually the resulting numerical problems would not allow the computer simulation software to run to completion.

As will be shown in the next section, the Q_{Fd} estimation technique is not responsive enough to handle the extreme maneuvers of an air-to-air missile, and other techniques appear to be more promising for adapting to detected maneuver initiations. This fact, combined with the problems just discussed, cause serious doubts as to the use of Q_{Fd} estimation. However, it should be pointed out that Q_{Fd} estimation

deserves a much more thorough study, especially for scenarios between the benign straight line trajectories and violent 20g turns, the two extremes employed in this study. Also, the robustness of a constant \underline{Q}_{Fd} needs to be tested.

Maneuver Required Modifications

The estimation of \underline{Q}_{Fd} adaptively, as described in Chapter IV, was intended to allow tracking of a maneuvering target while minimizing position errors on all trajectories. The first runs of the eight state filter on trajectory 3 (20g pullup) were disappointing. Several problems and potential solutions surfaced and are discussed in the following paragraphs.

The FOV of the 8 x 8 array (160 μ rads x 160 μ rads) was too small for highly maneuvering targets. The 20g lateral acceleration can cause the image to leave the FOV in three samples (.1 second) if the position update is not accurate. A related difficulty is resolution (number of pixels per FOV dimension). Using an image size of 5 pixels by 1 pixel ($\sigma_{vT} = 5$, $\sigma_{pvT} = 1$), virtually all the position information for the σ_{pv} direction is contained within 1 to 2 pixels of the peak position. That is, the high gradients occur within that region. The 20g acceleration can move the image 3 to 4 pixels laterally in two sample periods. Figure 42 shows this idea. With this much separation between the image peak intensity position and the filter's estimate of that position, the linearization used in the update

$$H = \left. \frac{\partial \underline{h}(\underline{x}, t)}{\partial \underline{x}} \right|_{\underline{x} = \hat{\underline{x}}(t_i^-)} \quad (149a)$$

is no longer valid. With significant separation, as in Figure 42, two factors cause the \underline{H} function to be in error. First, as the peaks of the intensity image and the filter \underline{h} function separate, the evaluation point ($\hat{\underline{x}}(t_i^-)$) is incorrect and the linearization of equation (149a) is in error.

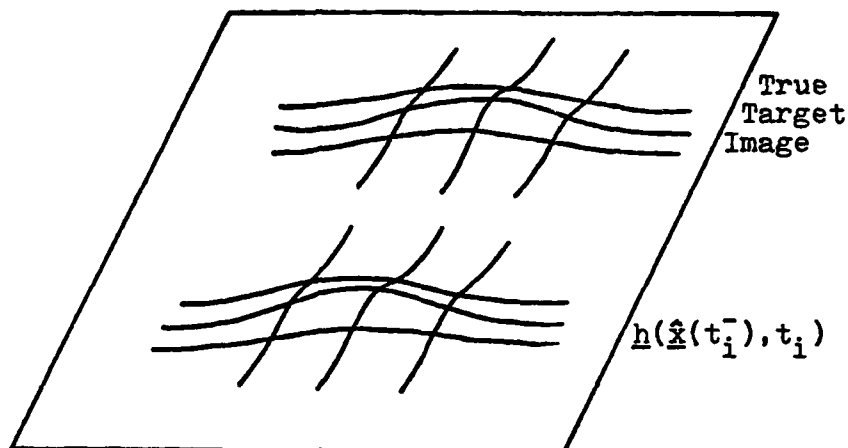


Figure 42. Image/Filter Image Separation by Acceleration

Second, the tails of a Gaussian intensity function are nearly constant and the partial derivatives of equation (149a) are approximately zero. These two effects combine to suppress any update when the situation of Figure 42 occurs. Even if the image is in the FOV, the tracker will probably lose the target, which makes detection and update imperative before one sample is past.

The \hat{Q}_{Fd} estimation is accomplished in a way such that $\hat{Q}_{Fd}(t_i)$ is used for the next sample period to time t_{i+1} . Thus, for the first sample after maneuver initiation at time t_i , \hat{Q}_{Fd} is still based on the (cross range) non-maneuvering portion of the trajectory. Therefore, the filter gains are low and the update is small even though the residuals are high. This response was insufficient to allow continued tracking and the image simply accelerated laterally out of the FOV. Moreover, on the benign portion of the trajectory, the \hat{Q}_{Fd} estimate had become small (as it should with little dynamic noise) and the fading memory smoother would not allow a rapid increase in entries of \hat{Q}_{Fd} .

This problem required several parameters and some elements of the state vector to be changed significantly in one sample period to maintain track. For example, the elevation acceleration value, x_6 in the state vector, must be changed from 0 to approximately 2400 pixels/sec² when the lateral (β) acceleration is applied. \hat{Q}_{Fd} and/or the P matrix entries need to be increased to avoid losing track while changing values in the state vector. Table XVII shows some useful variables that could indicate and quantify a maneuver. The quantity $\Delta x^T \Delta x$ is used to indicate a maneuver where

$$\Delta x^T \Delta x = \text{tr}(\Delta x \Delta x^T) \quad (150)$$

and

$$\Delta x = P(t_i^+) H^T(t_i) R^{-1}(t_i) [\Gamma(t_i) - h(\hat{x}(t_i^-), t_i)] \quad (150a)$$

Equation (150) was chosen as the maneuver indicator because $\Delta x \Delta x^T$ was already available from the \hat{Q}_{Fd} estimation algorithm. In many applications the residuals, in one form or another, are used directly to monitor filter performance. But, in this case the residual vector consists of 64 elements and to use them is computationally impractical. A comparison of $\Delta x^T \Delta x$, the azimuth and elevation elements of $H^T(\Gamma - h)$ and true errors for time starting at 3.9 seconds is given in Table XVII. Since $\Delta x^T \Delta x$ is used for detection, Δx would seem appropriate to quantify a maneuver. The difficulty with using Δx is that equation (150a) contains $P(t_i^+)$ which might have been artificially changed (protection from negative diagonal elements) and possibly one or more elements of Δx could be zero. To avoid this possibility and the computational burden of forming the residual vector inner product, $H^T(\Gamma - h)$, a six element vector, is used to quantify the maneuver. With R being a diagonal matrix, $H^T(\Gamma - h)$ is already computed in the filter and requires no new computations. Maneuver initiation occurs at 4.0 seconds and shows up in the filter at 4.03 seconds.

TABLE XVII
Maneuver Indicators

Time	Prospective Indicators			Actual Performance	
	$\Delta \underline{x}^T \Delta \underline{x}$	$H^T(\Gamma-h)$ (azimuth element)	$H^T(\Gamma-h)$ (elevation element)	x_{peak} azimuth error δ_α	y_{peak} elevation error δ_β
3.90	.1024	-23.0	743.6	-.17	.17
3.93	.0082	-11.1	30.3	-.19	.05
3.97	.0063	-11.1	-110.2	-.01	.01
4.0	.0376	- 6.6	-575.4	-.07	-.19
maneuver initiation					
4.03	.9263	28.2	3105.0	-.07	1.59
4.07	.0077	2.8	39.0	.18	5.37
4.10	.0181	-1.5	-11.4	.37	12.11

Note the increase in $\Delta \underline{x}^T \Delta \underline{x}$ and the elevation position element of $H^T(\Gamma-h)$ at time 4.03 seconds. The sign of the $H^T(\Gamma-h)$ element tells the direction of the maneuver, in this case the acceleration is up (positive) perpendicular to the velocity vector. The magnitude of 3105.0 is related to how far apart the image and filter function h centers are. Note that although the upward acceleration continues, the indication that a maneuver has occurred is lost by 4.07 seconds. Therefore, it is imperative that the filter detect the maneuver immediately and perform the appropriate update at 4.03 seconds with that knowledge.

Upon maneuver detection three actions should be taken. At least an approximation or crude estimate of acceleration that can describe the maneuver must be provided to the

filter. The filter gains must be increased directly (rather than allow them to build slowly by estimating Q_{Fd} with resulting P increase) to force increased emphasis of measured data over filter predicted location, damp out any transients and account for uncertainties in maneuver quantification. Also, the FOV should be increased to avoid letting the image escape the FOV. Taking the above steps should allow the filter to track the target after initiation of a maneuver. Expanding the FOV was not done in this research for reasons mentioned in Chapter II.

Looking at Table XVII there is a definite correlation between δ_α , δ_β and the respective elements of $H^T(\Gamma-h)$. This correlation can be exploited to quantify δ_α and δ_β . If δ_α and δ_β are then assumed to have been the result of a discontinuity in the acceleration, an estimate of this change in acceleration can be computed.

For $|\delta_\alpha| \approx |\delta_\beta|$ of Table XVII, the respective elements of $H^T(\Gamma-h)$ differ considerably. $H^T(\Gamma-h)$ is given by

$$H^T(\Gamma-h) = \left. \frac{\partial h(\underline{x}(t_i), t_i)}{\partial \underline{x}} \right|_{\underline{x} = \hat{\underline{x}}(t_i^-)} [\Gamma(t_i) - h(\hat{\underline{x}}(t_i), t_i)] \quad (151)$$

which is a strong nonlinear function of σ_v , σ_{pv} and θ (see eqn. (130)). Because $\sigma_{vT} \approx 5$ and $\sigma_{pvT} \approx 1$ significant differences will exist in the associated $|H^T(\Gamma-h)|$ values. Notwithstanding, there is a definite relationship between the position errors, $|\delta_\alpha|$ and $|\delta_\beta|$, and their respective $H^T(\Gamma-h)$ elements.

An analytical solution for δ_α and δ_β is not possible so an empirical approach was attempted to obtain this quantitative relationship. Two coordinate frames seem reasonable to approach this problem. The filter uses the azimuth/elevation LOS system, but lateral acceleration is expressed most easily in a missile oriented frame (σ_v , σ_{pv}). The filter frame is desirable insofar as the result is used in

that frame. To this end, the size, shape and orientation parameters of the filter (σ_{VF} , σ_{pVF} , $\sin\theta$, $\cos\theta$) are transformed into σ_{az} and σ_{el} . This is done using the following equations.

$$\sigma_{az} = \left[\frac{\sigma_{pVF}^2}{1 - e^2 \cos^2 \theta} \right]^{1/2} \quad (152)$$

$$\sigma_{el} = \left[\frac{\sigma_{pVF}^2}{1 - e^2 \sin^2 \theta} \right]^{1/2} \quad (153)$$

where

$$e = \left[1 - \frac{1}{AR_F^2} \right]^{1/2} \quad (154)$$

and subscripts az and el stand for azimuth and elevation, respectively.

The functional relationship $g(\cdot)$ desired, for example δ_β , is of the form

$$\delta_\beta = g\{[\underline{H}^T(\underline{\Gamma}-\underline{h})], \sigma_{az}, \sigma_{el}\} \quad (155)$$

where σ_{az} = image dispersion in the azimuth direction

σ_{el} = image dispersion in the elevation direction

The function $g(\cdot)$ is strictly a function of the entire $\underline{H}^T(\underline{\Gamma}-\underline{h})$ vector. However, if tracking correctly, the dependence on all elements except the one associated with elevation (β) can be neglected.

Because of nonlinearities and other considerations, plots of $\log|\delta_\beta|$ vs $\log|[\underline{H}^T(\underline{\Gamma}-\underline{h})]_\beta|$ were made. Figure 43 shows one such plot for a case with $AR_T = 3$. For values of $|\delta| > .1$ the high correlation with $\underline{H}^T(\underline{\Gamma}-\underline{h})$ is preserved. For values of $|\delta| < .1$ the data is noisy and the correlation

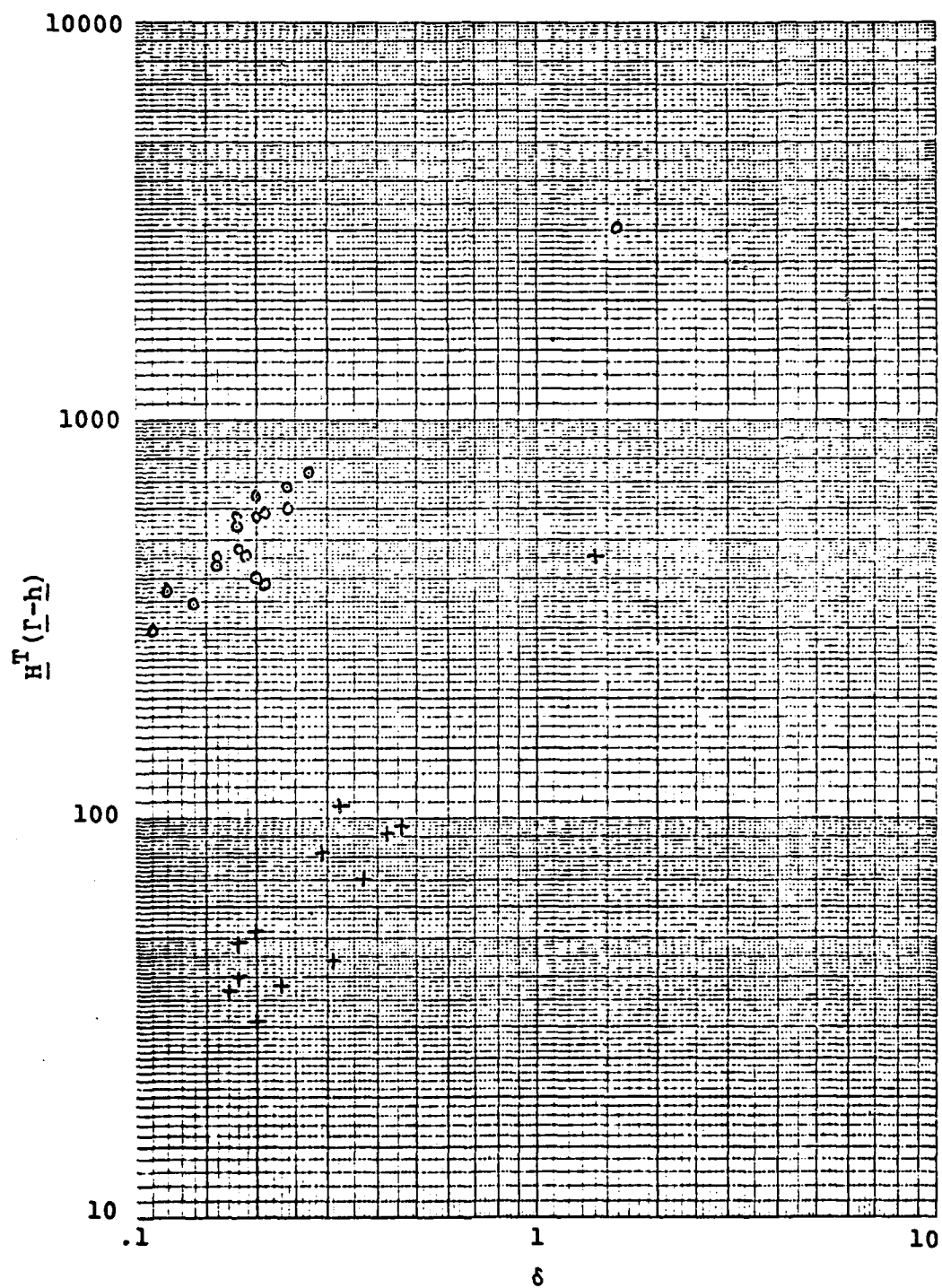


Figure 43. $H^T(\Gamma-h)$ vs δ

is lost. The function sought from these graphs was of the following assumed form

$$\log |[\underline{H}^T(\underline{\Gamma}-\underline{h})]_{\rho}| = a \log |\delta_{\beta}| + b \quad (156)$$

where a and b are the adjustable parameters. In both cases ($AR_T = 5$ and $AR_T = 3$), a was approximately constant, within the resolution of the measurements, and set equal to .82. Figure 44 shows the intercepts (b) for the two cases and, as can be seen, b is nearly a linear function of σ .

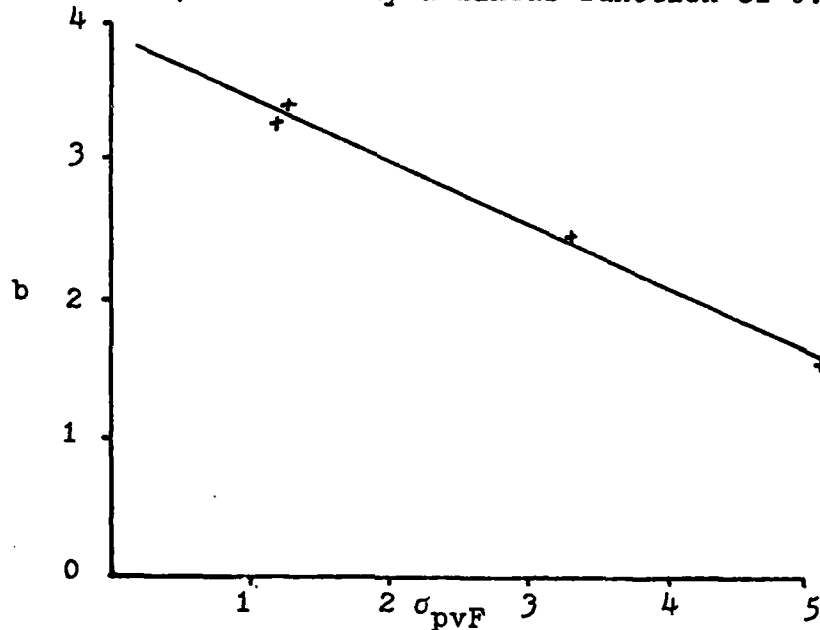


Figure 44. Intercept for Equation (156)

Equation (156) then becomes

$$\log |[\underline{H}^T(\underline{\Gamma}-\underline{h})]_{\beta}| = .82 \log |\delta_{\beta}| - .46 \sigma_{e1} + 3.9 \quad (157)$$

Solving for $|\delta_{\beta}|$

$$|\delta_{\beta}| = \exp \left\{ 2.303 \left[\frac{\log |[\underline{H}^T(\underline{\Gamma}-\underline{h})]_{\beta}| + .46 \sigma_{e1} - 3.9}{.82} \right] \right\} \quad (158)$$

This equation is applied to the elevation direction and a similar equation applied to the azimuth direction when a maneuver is detected by $\Delta \underline{x}^T \Delta \underline{x}$ (or $\underline{H}^T (\underline{\Gamma} - \underline{h})$ entries) surpassing a threshold. The δ 's are assumed to have resulted from an acceleration not reflected by the acceleration state estimates. In equation form

$$\delta_{\beta} = \frac{1}{2}(\Delta a_{\beta})\Delta t^2 \quad (159)$$

is solved for Δa_{β} and the sign of $[\underline{H}^T (\underline{\Gamma} - \underline{h})]_{\beta}$ incorporated to give

$$\Delta a_{\beta} = \frac{2|\delta_{\beta}|}{\Delta t^2} \cdot \text{sgn}\{[\underline{H}^T (\underline{\Gamma} - \underline{h})]_{\beta}\} \quad (160)$$

where $|\delta_{\beta}|$ is estimated using equation (158). Assuming the maneuver was detected at t_i , the incremental changes in acceleration (Δa_{α} , Δa_{β}) are added to the acceleration state estimates and the modified state vector set equal to $\hat{\underline{x}}(t_{i-1}^+)$. This new state vector is propagated from $\hat{\underline{x}}(t_{i-1}^+)$ to $\hat{\underline{x}}(t_i^-)$ and used as usual.

Incorporating this algorithm into the filter and running some test cases showed promising results. The filter would track for several samples beyond the point of loss of lock in the initial design without the estimated acceleration incorporated. The acceleration obtained from equation (160) is not sufficiently accurate to allow continued tracking without good updates in the next few samples. Also, the acceleration is not constant in the azimuth, elevation, LOS coordinate frame. Thus, the updates in the filter must be accurate and rapid. Table XVIII shows a typical sample result when the acceleration estimate is computed and incorporated into the state vector but the gains are left unchanged. Comparing Table XVIII to Table XVII shows this improved performance.

TABLE XVIII
Elevation Acceleration Estimate Only

Time	δ_{α}	δ_{β}	Remarks
3.9	.22	.17	maneuver initiated
3.93	.20	.03	
3.97	.37	-.02	
4.0	.31	-.20	
4.03	.31	.23	
4.07	.55	.18	
4.1	.68	.08	
4.13	1.69	.00	
4.17	2.99	-.37	
4.20	4.58	-.87	
4.23	6.86	-1.15	

To increase the filter gains, the P matrix was redefined by increasing the variances of the acceleration and velocity components. For velocity uncertainty the variance was increased to 100 and for acceleration uncertainty the variance was increased to 300. For implementation these variances would be a function of the accuracy of the maneuver detection algorithm and would be similar to an acquisition mode. The performance of the filter improved and it was able to track the image for about .5 seconds. Table XIX shows the result when the increase in elements of P is added. This sample corresponds directly to that of Table XVIII (same samples of noise process used in simulation). The maneuver detection, Δa computation and P increase function properly. However, at 4.2 seconds two diagonal elements of the P matrix become negative and the affected rows and columns were set to zero. This seemed to cause the algorithm

TABLE XIX

Elevation Acceleration Estimate and Increased P Matrix

Time	δ_α	δ_β	Remarks
3.90	.22	.16	
3.93	.20	.03	
3.97	.38	.03	
4.00	.31	-.21	Maneuver initiated
4.03	.31	1.59	First \hat{x} (4.03 ⁻) (unmodified)
4.03	-.46	.2	New \hat{x} (4.03 ⁻) (modified) true acceleration was 2448 pixels/sec ² estimated value was
4.07	-1.68	-.34	
4.10	-1.87	-.35	
4.13	-2.38	-.13	
4.17	-2.18	.34	
4.20	-1.59	.25	Two elements of <u>P</u> (4.20 ⁺) < 0. Rows and columns set to 0. No x_4 or x_6 update.
4.23	-1.05	.38	
4.27	.15	-.06	Element of <u>P</u> (4.27 ⁺) < 0. Row and column set to 0. No x_4 update.
4.30	1.29	-.45	The δ_α tripped maneuver detector and was corrected
4.30	.78	-.74	New \hat{x} (4.30 ⁻)
4.33	.32	-1.56	
4.37	.57	-1.36	
4.40	.17	-1.15	
4.43	-.17	-.63	
4.47	.09	-.90	
4.50	.73	-1.40	
4.53	1.49	-1.69	
4.57	1.47	-2.33	
4.60	1.86	-2.99	Filter function <u>h</u> and image no longer overlap significantly (lost track)

to become unstable. Note the increased error in the azimuth direction (acceleration was in the elevation direction). This occurred because the Δa_α was computed from a $|\delta_\alpha|$ and $|\underline{H}^T(\underline{r}-\underline{h})|_\alpha$ with insufficient correlation. Even so, the filter was recovering from the erroneous Δa_α until the \underline{P} matrix becomes indefinite at 4.2 seconds. This can be remedied in part by lower bounding the \underline{P} diagonal terms by a positive number (.1 for this research). Continued work in this area is warranted, including the incorporation of FOV expansion and contraction.

Final Filter Performance

The final filter algorithm included I_{\max} , AR and σ_v estimation, fixed \underline{Q}_{Fd} matrix, and an acquisition scheme all incorporated into the eight state formulation. Recall in Chapter IV the desire to use the six state filter vs. the eight state formulation. Because of the non-inertial accelerations the rotating LOS coordinate system presented, the eight state filter was selected to reduce these effects.

The \underline{Q}_{Fd} adaptive estimation proved to be inadequate. Time constraints prevented a thorough investigation of the causes for its trouble, but it seemed to create more problems than it solved. It may be useful for dynamic environments below maneuver detection threshold, but its performance benefit would need to be compared to the cost in computer time and potential induced stability problems before it could be recommended for implementation. So the final filter uses a \underline{Q}_{Fd} high enough to allow acquisition and is fixed at that value.

Performance of this filter on trajectory 2 (cross range) is shown in several figures starting with Figure 45. This figure shows the azimuth mean position error plus or minus one sigma. Comparing this to Figure 32 shows that the eight state filter exhibits little evidence of the non-inertial effects. The maximum mean error is .175 pixels

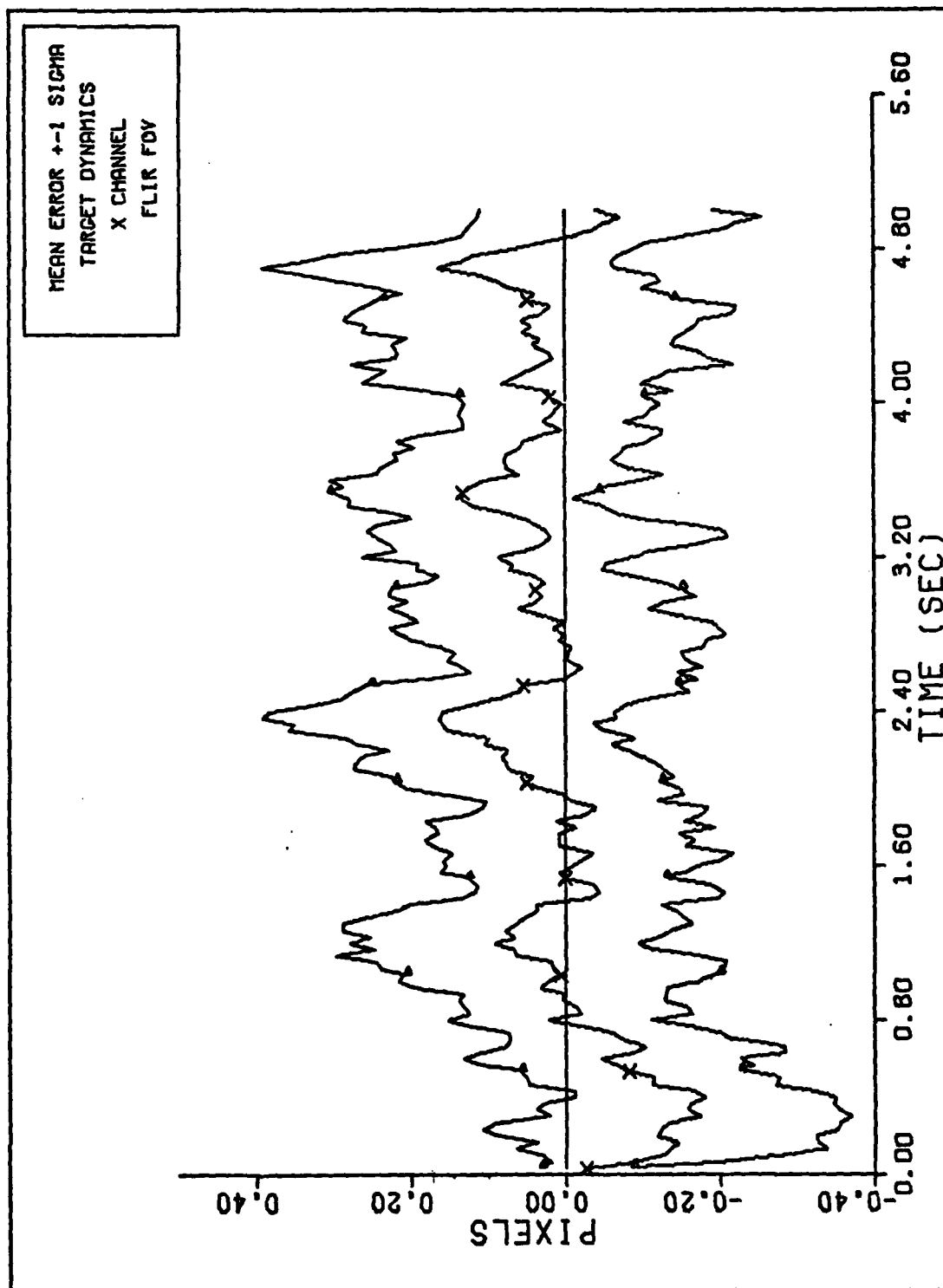


Figure 45. X CHANNEL DYNAMICS ERROR (S/N=12.5)
Case 36: Target Position Mean Error and
1 Sigma Envelope - X Channel

(3.5 μ rads) and the three sigma value for this point is .84 pixels (16.8 μ rads). Figure 46 shows the azimuth mean velocity error to less than .17% of the velocity value. Generally, acceleration magnitude will be nonzero at hand-off and there will be a transient before the acceleration estimate is accurate. This transient is damped out by the acquisition mode with its higher P elements reflecting the uncertainty at hand-off.

Figure 47 shows the filter estimate of the azimuth position standard deviation vs. the actual sample standard deviation from the simulation. Figure 48 shows the corresponding results for azimuth velocity. Both plots show a slight overestimation of the variance, but nevertheless good performance. The dynamic driving noise strength ($\sigma_D^2 = 600$) was tuned to this trajectory (2) and the filter should perform well. However, limited time prevented an extensive investigation of suitable noise strengths. A more dynamic environment would make a higher strength noise appropriate. If good performance required a well-matched Q_{Fd} then adaptive estimation of Q_{Fd} could be used to advantage. Figure 49 through Figure 52, for the elevation direction, are similar to the plots for azimuth. The same comments given above are applicable to the elevation direction. Parameter values for this run are given in Appendix K case 36.

For the highly dynamic environment of trajectory 3 (20g pullup), partial results are given in the preceding sections of this chapter. The Monte Carlo runs were not completely successful, so no corresponding plots were made.

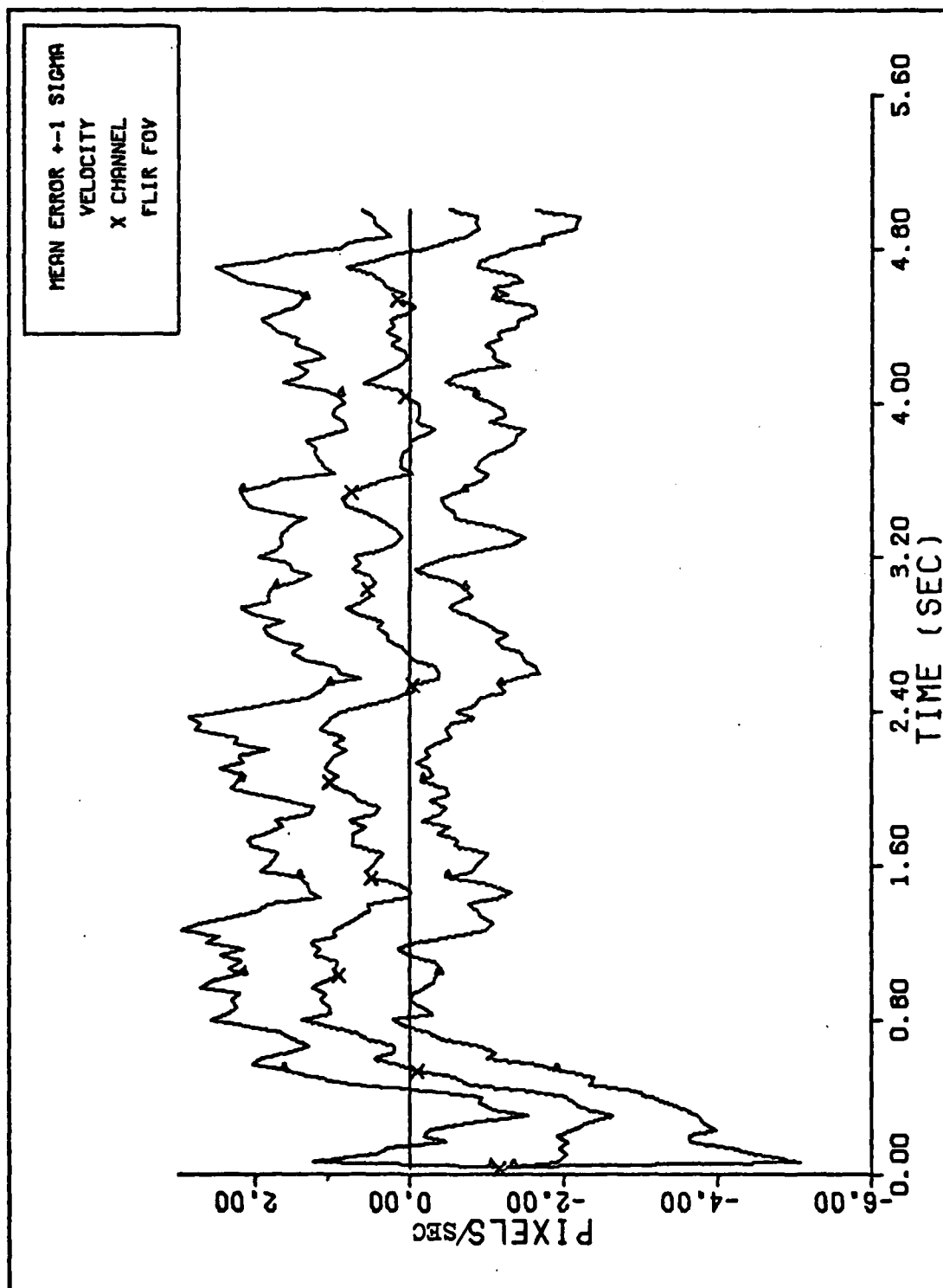
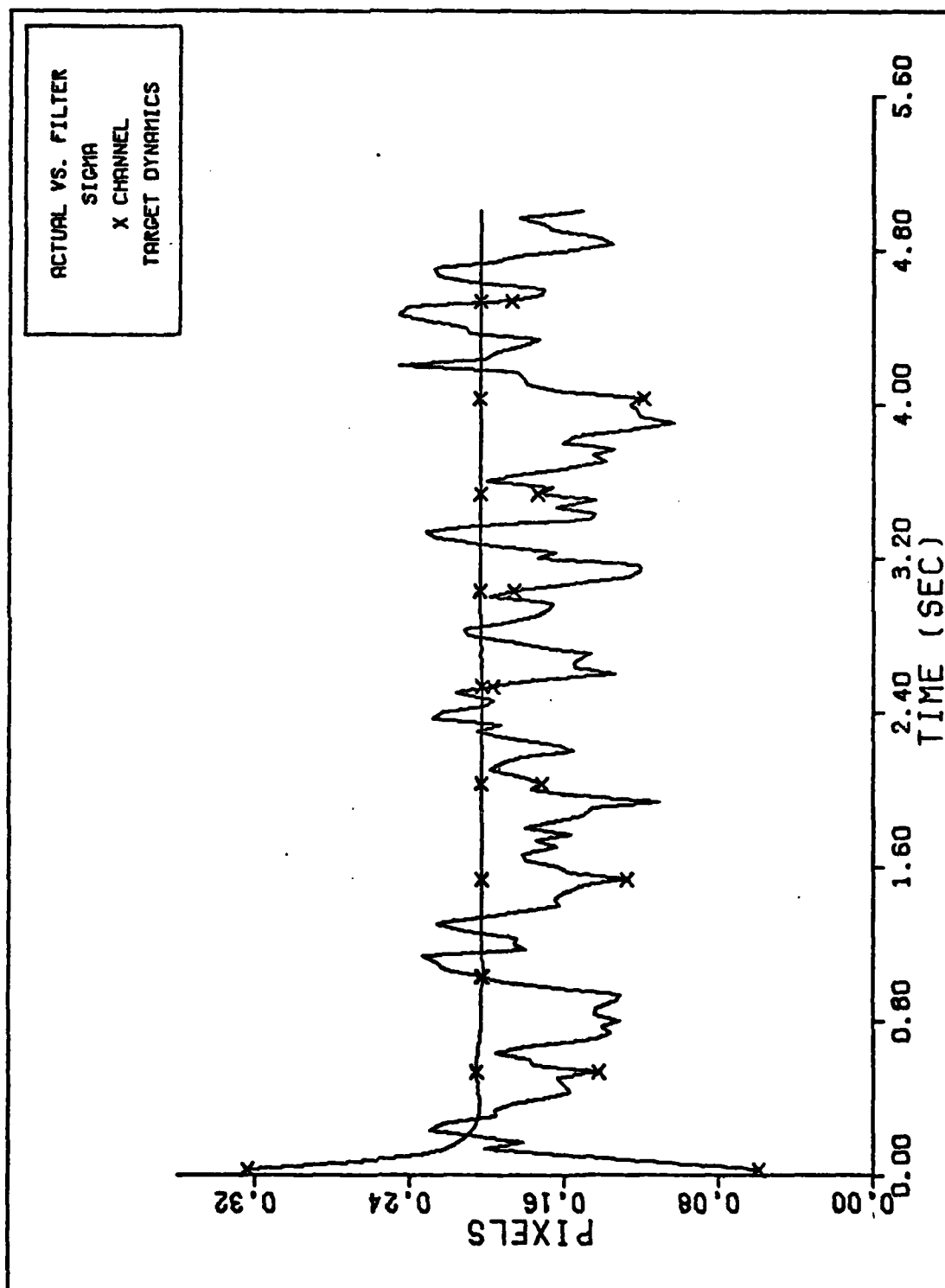


Figure 46. Case 36: Target Velocity Mean Error and 1 Sigma Envelope - X Channel



FILTER VS. ACTUAL SIGMA PLOT (S/N = 12.5)

Figure 47. Case 36: Target Position Standard Deviation, Actual and Filter-Indicated - X Channel

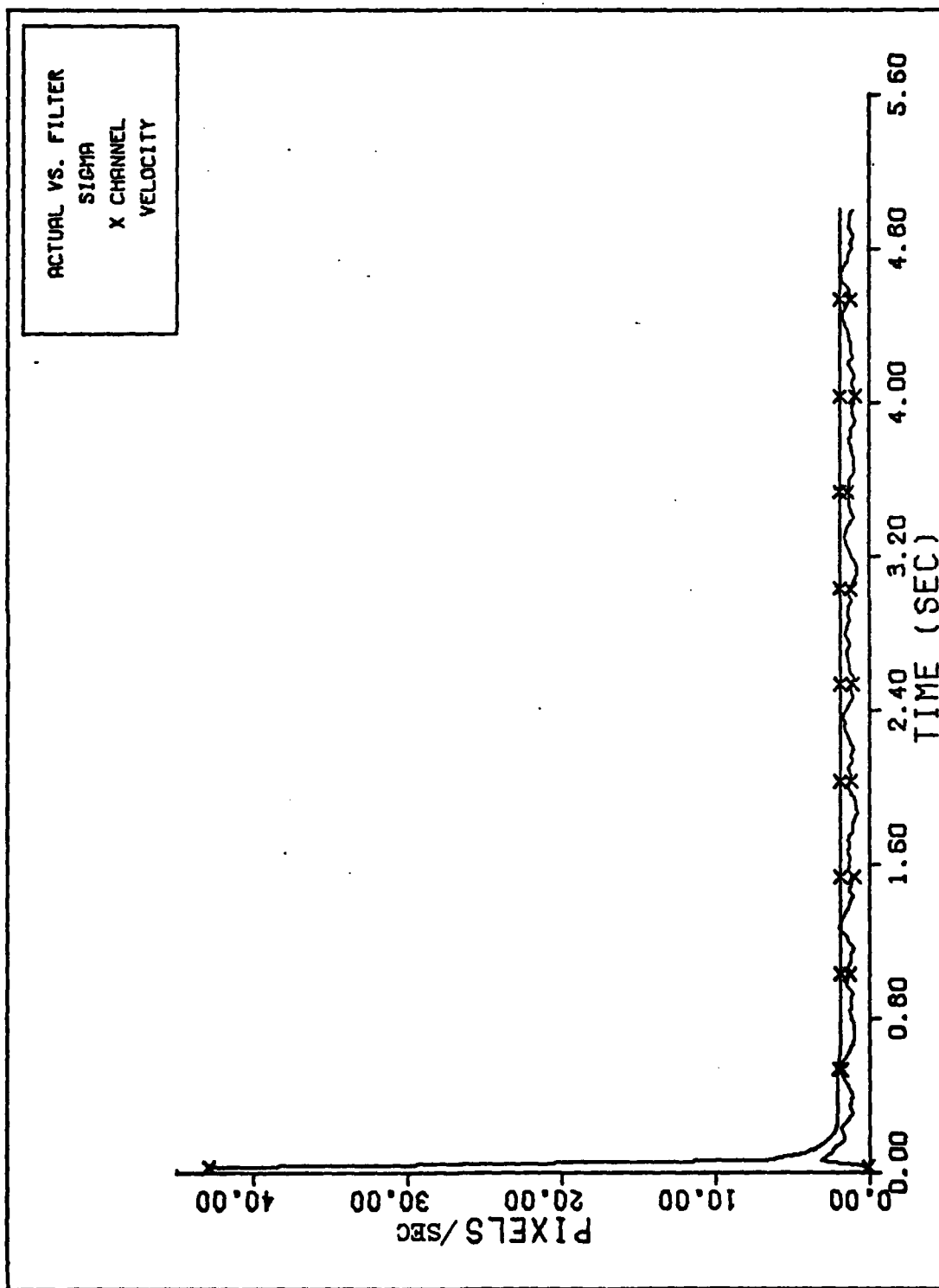
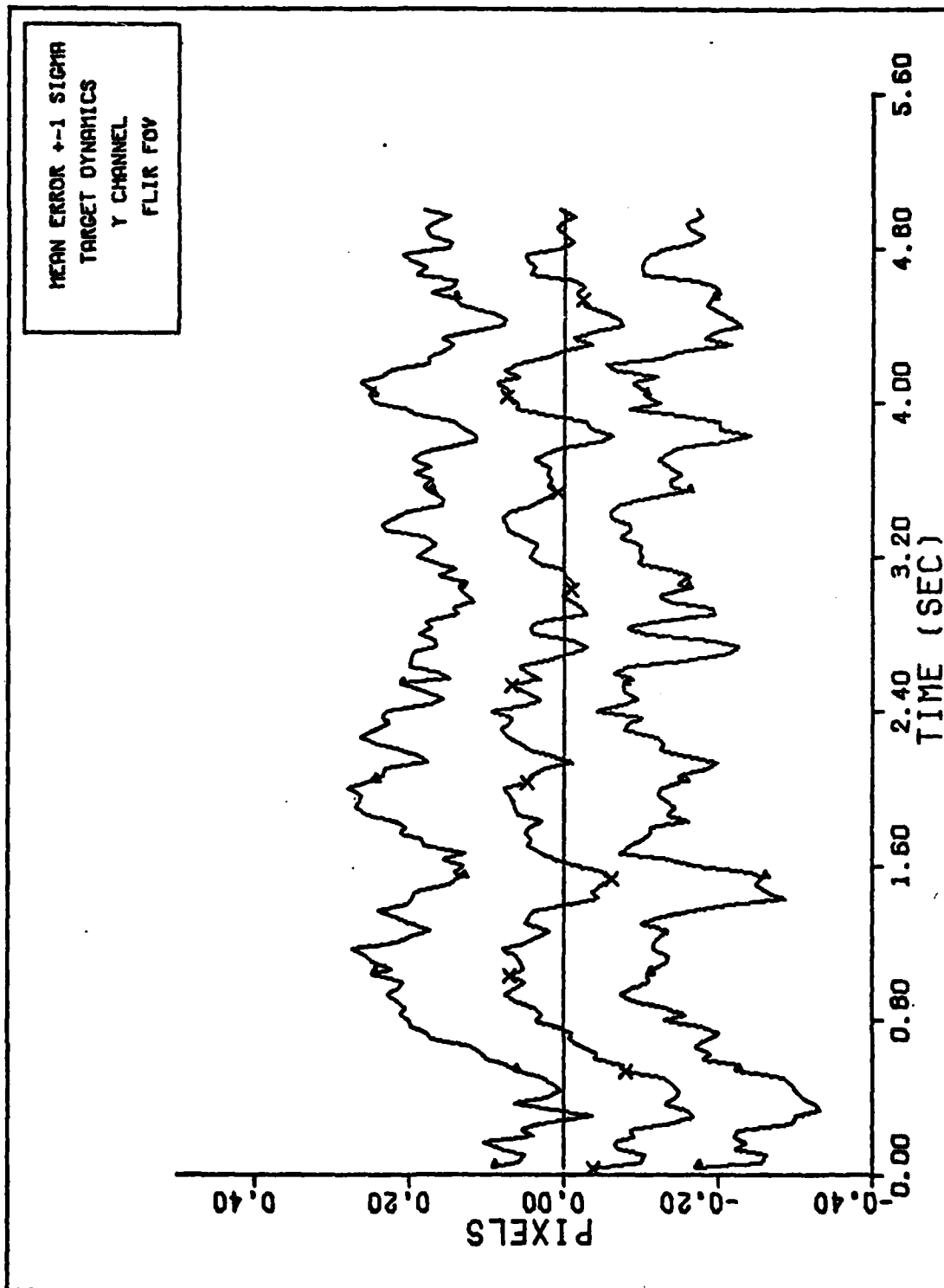
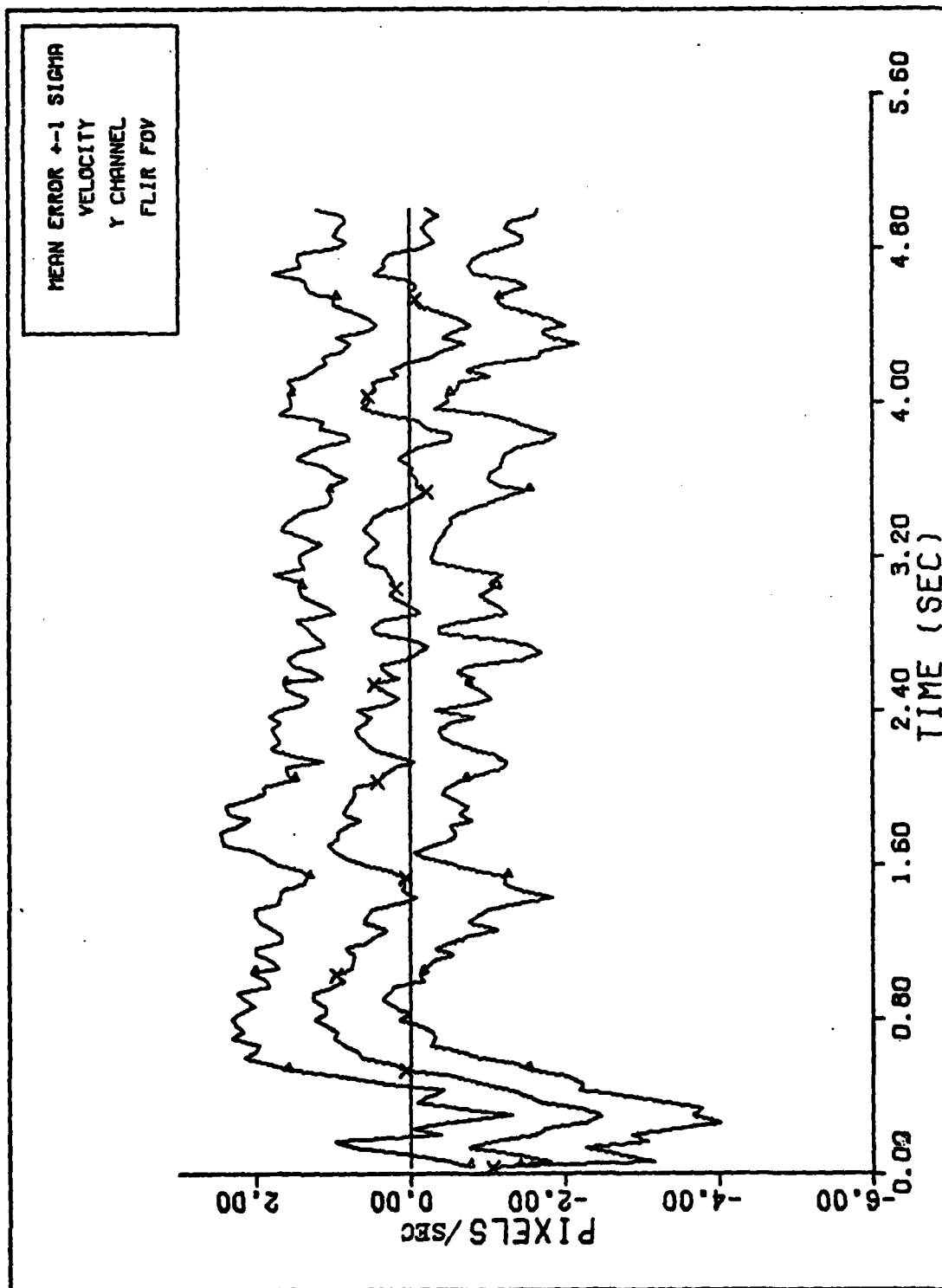


Figure 48. FILTER VS. ACTUAL SIGMA PLOT ($S/N = 12.5$)
Case 36: Target Velocity Standard Deviation,
Actual and Filter-Indicated - X Channel



Y CHANNEL DYNAMICS ERROR (S/N=12.5)
Figure 49. Case 36: Target Position Mean Error and 1
Sigma Envelope - Y Channel
142



Y CHANNEL VELOCITY ERROR (S/N=12.5)

Figure 50. Case 36: Target Velocity Mean Error and 1 Sigma Envelope - Y Channel

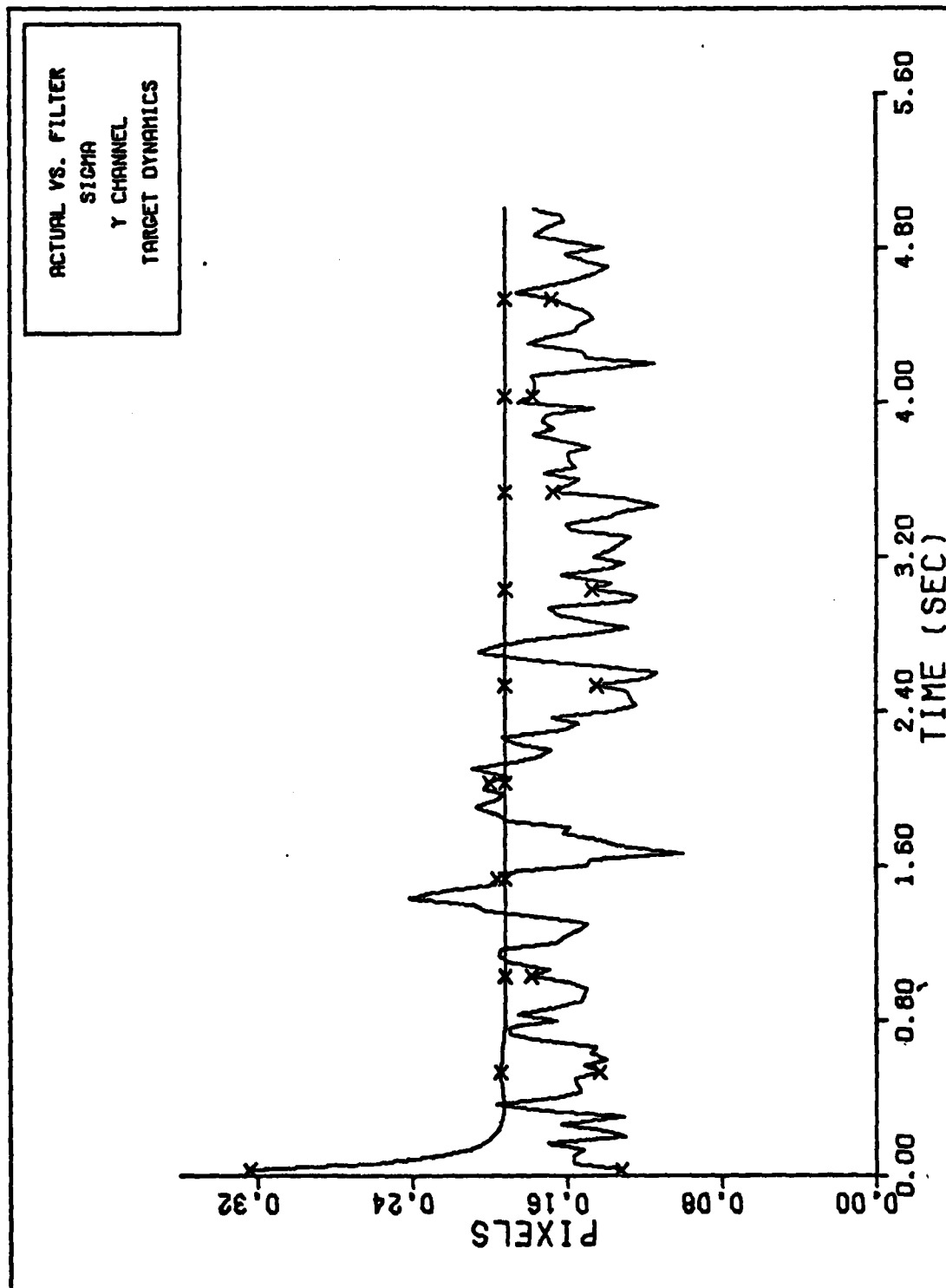


Figure 51. FILTER VS. ACTUAL SIGMA PLOT (S/N = 12.5)
Case 36: Target Position Standard Deviation,
Actual and Filter-Indicated - Y Channel

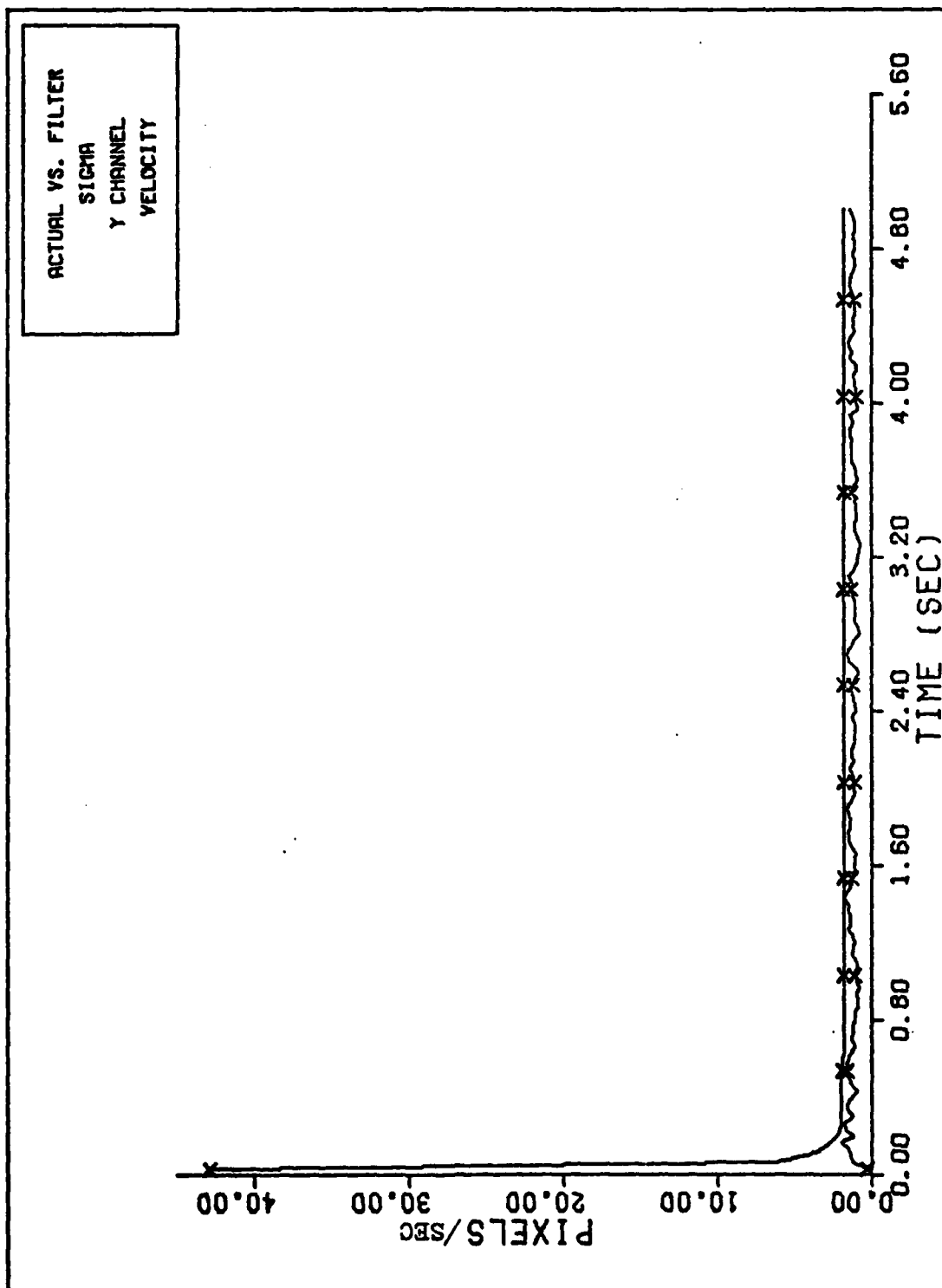


Figure 52. FILTER VS. ACTUAL SIGMA PLOT (S/N =12.5)
Case 36: Target Velocity Standard Deviation,
Actual and Filter-Indicated - Y Channel

VI. Conclusions and Recommendations

Conclusions

The scope of this study allowed numerous conclusions, particularly concerning the various filter techniques tested. Most of the filter techniques, such as adaptive schemes and state space formulations, which are addressed in these concluding remarks, were formulated in detail in Chapter IV and analyzed in Chapter V.

The fairly simple techniques developed for the estimation of the maximum target intensity, I_{\max} , achieved very good accuracy. With a true $I_{\max} = 25$, the estimation process averaged 24.99 with a standard deviation of .299. The adaptive process was specifically tuned to this level of target intensity, but serious degradation is not anticipated at other target intensities using this estimation scheme. And, the small price in computational load is readily affordable to identify this critical parameter accurately.

Similarly, the simple gradient schemes devised for estimating the size of the target in its longitudinal axis, σ_v , and the aspect ratio of the target, AR, were sufficiently accurate and easily implemented. These estimation schemes performed so well that further investigations into more precise approximations to solutions of the full Maximum Likelihood Equations were not conducted. For a true AR of 5, the estimation scheme averaged 4.07 with a standard deviation of .16. The large bias between the true and estimated values is well understood, and very accurate target modelling and state estimation are still achieved as is discussed in detail in Chapter V. For a true σ_v of 5, the estimation of σ_v averaged 4.83 with a standard deviation of .32. Again, the slight bias is understood and causes no problems in target modelling.

The eight state filter is preferred to the six state filter to provide more accurate tracking of the missile as

can be seen by comparing Figures 32 through 35 (six state filter) to Figures 40 and 41 (eight state filter) in Chapter V. The eight state filter is needed to accommodate the non-inertial accelerations caused by the rotating FLIR reference frame as seen in the straight line flight test cases. Also, the acceleration estimation scheme developed for high g turns is more easily used in conjunction with the eight state filter because x and y channel accelerations are two of its states.

The technique for estimating the dynamic driving noises, \underline{Q}_{Fd} , was not conclusively determined to be useful. Much was learned concerning procedures to combat apparent stability problems that occur when using the \underline{Q}_{Fd} estimation process. Zero lower bounding of \underline{P} and \underline{Q}_{Fd} , and upper bounding on \underline{Q}_{Fd} were not effective solutions to the instability of the filter which appeared as negative values on the main diagonal of \underline{P} . A brief study of positive lower bounding of \underline{P} and \underline{Q}_{Fd} demonstrated good potential for solving the stability problems noted. It was discovered that \underline{Q}_{Fd} estimation was not fast enough for hard maneuvers but it could be used perhaps for intermediate (medium maneuvers) cases which were not investigated in Chapter V.

The maneuver detection and acceleration estimation techniques showed great promise in leading to the ability to track the missile in high g turns with a very limited field of view. The maneuver detection technique is based on large changes in the scalar $\Delta \underline{x}^T \Delta \underline{x}$, where $\Delta \underline{x}$ equals the gain, \underline{K} , times the residuals, or on large changes in $\underline{H}^T [\underline{\Gamma} - \underline{h}]$ (note $\underline{K} = \underline{P}^+ \underline{H}^T \underline{R}^{-1}$ and $\underline{R} = \underline{R} \underline{I}$ here). These techniques successfully detected all the maneuvers tested which were extreme. The acceleration estimate which was based on the linearized measurement matrix multiplied by the residuals, $\underline{H}^T (\underline{\Gamma} - \underline{h})$ proved capable of identifying the acceleration of a high g turn to within five percent of its true value in one sample period. Immediately after maneuver detection two actions were taken: \underline{P}^+ was increased which immediately

increased the gain ($K = P^+ H^T R^{-1}$); and an acceleration estimate, as described above and in Chapter V, was incorporated into the filter. These procedures were effective for about half a second, but the filter then lost track of the target. The loss of track was not directly attributable to the initial actions taken, but was caused by the filter's inability to keep the gains high enough throughout the turn. A procedure, which was not evaluated but could be effective in tracking missiles in g turns, would additionally expand the field of view of the filter by summing up pixel values to represent a single measurement, and is described in the recommendations section to follow.

Finally, in the high levels of S/N (10 or greater) studied, the spatial and temporal correlations of the background noise were not found to be significant in their effects on the filters performance, as was shown in Chapter III. They were found to have a significant impact at S/N equal to two or lower. The combination of spatially correlated noise and temporally correlated noise was found to have more effect than each one taken separately as can be seen in the cases presented in Chapter III.

Recommendations

Many areas needing more extensive study were bypassed in an effort to progress the general design of an effective filter as far as possible. Hopefully, this study has solved some problems and provided useful insights towards a final adaptive extended Kalman filter. Future research on this problem can go in many directions.

More study needs to go towards testing the final filter design to determine more completely its capabilities and to see if it can track a turning missile with minor modifications. In particular, investigations of target scenarios between the straight missile flight path and the 20 g turn are needed. Also, the maneuver detection and acceleration estimation schemes will require more study.

Some major modifications could include (but are not limited to) alternate reference frames for problem formulation, alternate target acceleration models, enlarging the field of view (dynamic pixel compression schemes), and dual parallel filters (a form of multiple model filtering, Ref 8: 10-103 to 10-106). Alternate reference frames could possibly lead to more accurate or more efficient filter formulations. For example, a vehicle fixed reference frame filter formulation could provide better filter performance during high g missile maneuvers. Alternate models for target acceleration, such as constant turn rate model or first order Markov process, could enhance filter performance as compared to the present model which represents acceleration as Brownian motion. The idea of enlarging the field of view is motivated by the need to keep a maneuvering target in the field of view considered by the filter. One such scheme would sum up 2-by-2 or 4-by-4 arrays of pixels into one measurement to allow an expansion of the field of view without changing the size of the measurement array in the filter. The dual filters idea may be most useful. One filter, such as designed here, could be used to track linear missile trajectories, and the other filter with high gains, possibly constant, would track the high g missile turns. Switching between filters could be provided by a maneuver detection scheme as studied in this research.

Another study could concentrate on the estimation technique for \underline{Q}_{Fd} in order to analyze and solve the stability problems noted in this research. The positive lower bounding of \underline{Q}_{Fd} and \underline{P} would be good starting points for such a study, but a greater variety of target scenarios should be investigated to establish more universal lower bounds for the diagonal terms of each matrix. Further research into \underline{Q}_{Fd} estimation techniques should address its role in target acquisition strategies, and a tradeoff study should be conducted to determine whether performance benefits warrant the additional complexity of the adaptation.

An idea which may merit investigation is the modelling of some targets by two or more related elliptical contour intensity distributions. Targets such as aircraft, may appear to be composed of more than one major source of infrared (IR) radiation when viewed by an IR sensor. For example, aircraft nose, wingtips, engines and engine exhaust are all possible sources of high IR radiation. More effective tracking schemes could possibly be formulated with multiple intensity patterns which maintain a consistent geometrical relationship to each other.

A more long range research topic could focus on an efficient on line implementation form of proposed filters, such as square root filter forms or U-D factorization forms (Ref 15: Chapter 7). The sample rate for any effective tracking system, such as the 30 Hz rate currently used, will have to be relatively fast. This will require extreme efficiency and accuracy of the on line formulation used, and optimization of the algorithm in this regard should be undertaken.

The very specialized problem of tracking at low S/N will eventually need to be confronted. As shown in the present research, a part of such future study should include the spatial and temporal nature of the background noise. Thus, part of the study would entail more extensive real data analysis to evaluate the temporal and spatial correlations involved more precisely and to formulate new models as needed. This possibly could be accomplished by incorporating the temporal and/or spatial noise concepts into the filter, and, then, evaluate filter performance against a realistic, more complex truth model such as stored test data.

Future filter designs to incorporate range and/or range rate data may provide the best approach to the missile tracking problem. Such measurements would allow direct computation of compensation terms for non-inertial acceleration effects, and would enhance estimation of range-dependent

parameters to describe target size effects and maneuver rate capabilities. The price of the extra sensors needed for the range and range rate data could easily be outweighed by the accuracy and simplification gained.

The ultimate goal would be the formulation of a sub-optimal closed loop controller design. Incorporating both laser system and tracking system dynamics into a near optimal feedback system (as via forced separation design, separating the generation of deterministic controllers and estimators, and then combining them) would be a very complex problem requiring knowledge in several specialized areas such as optics, optics control, estimation theory, control theory, etc. This would be a very demanding task, directly exploiting the tracking filter designed in this research and extended as discussed above.

Computer Support

The pace and extent of this research effort was significantly retarded by the computer support available at AFIT. An estimated 30 percent more work could have been accomplished with a more reliable and responsive system. The authors encourage any steps to make AFIT's computer support better. Possible recommendations include more hardware support (i.e. it is unbelievable that AFIT has only one plotter), more personnel support, and AFIT's own dedicated computer.

Bibliography

1. Ulsamer, Edgar. "Defense Technology: Moving into Space," Air Force Magazine, June 1979.
2. Day, Bonner. "Progress on Energy Beam Weapons," Air Force Magazine, July 1978.
3. Ulsamer, Edgar. "USAF's R&D, Lean and Healthy," Air Force Magazine, December 1978.
4. Maybeck, Peter S. and Mercier, Daniel E. "A Target Tracker Using Spatially Distributed Infrared Measurements," Proceedings of the IEEE Conference on Decision and Control, Fort Lauderdale, Fla., December 1979.
5. Mercier, Daniel E. "An Extended Kalman Filter for Use in a Shared Aperture Medium Range Tracker," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1978.
6. Advanced Adaptive Optics Control Techniques, TR-996-1. The Analytic Sciences Corporation; prepared for the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, 6 January 1978.
7. Hogge, C. B. and Butts, R. R. "Frequency Spectra for the Geometric Representation of Wavefront Distortions Due to Atmospheric Turbulence," IEEE Transactions on Antennae and Propagation, Vol. AP-24, No. 2, March 1976. (Program supplied by authors.)
8. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 1. Academic Press, New York, N. Y., 1979.
9. Lewantowicz, Z. H., Captain, Air Force Weapons Laboratory, Kirtland AFB, New Mexico. Telephone discussion with Dr. P. S. Maybeck, 1 September 1979.
10. Maybeck, P. S. Stochastic Estimation and Control, Part II (Course notes from EE 7.66). Wright-Patterson AFB, Ohio, Air Force Institute of Technology, February 1975.
11. CRC Standard Mathematical Tables. Samuel M. Selbly, editor-in-chief. Chemical Rubber Co., Cleveland, Ohio, 1969.
12. Westinghouse Defense and Electronic Systems Center. Digital Correlation Tracker, Phase I Computer Simulations, TR-74-170. Prepared for the Air Force Weapons Laboratory, Kirtland AFB, New Mexico, March 1975.

13. Westinghouse Defense and Electronic Systems Center. User's Manual for the Computer Simulation Package Digital Correlation Tracker, F29601-74-C-0024. Prepared for the Air Force Weapons Laboratory, Kirtland AFB, New Mexico, July 1976.
14. Merritt, Paul H. Beam Intensity Calculations for Jittered Beams, TR-78-174. Prepared for the Air Force Weapons Laboratory, Kirtland AFB, New Mexico, 1978.
15. Fosha, Charles E., Leatham, Anthony L., and Neeland, Roger P. "The ASCOT Electro-Optical Sensor Employed as a Line of Sight Tracker with a Digital Estimator in Air-to-Air Engagements," Department of Astronautics and Computer Science, U. S. Air Force Academy, Colorado, 13 May 1977.
16. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 2. Unpublished text. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1978.
17. Firefly III System Mechanization, GE Document ACS-11,935. General Electric Aircraft Equipment Division, Binghamton, New York, 12 July 1979.

Vita

Robert L. Jensen was born on August 17, 1947 in San Diego, California. He graduated from San Diego State University in February 1972 and entered the Air Force that same month. Upon graduation from navigator training at Mather AFB, California in February 1973, he was assigned as a C-5A navigator at Dover AFB, Delaware. In May 1978, Captain Jensen was assigned to the Air Force Institute of Technology to study for a Masters Degree in Astronautical Engineering.

Permanent address: 3303 54th Street
San Diego, California 92105

Douglas Alan Harnly was born on February 25, 1952 at Ramey AFB, Puerto Rico. He graduated from Billy Mitchell High School in Colorado Springs, Colorado in June 1970 and entered the United States Air Force Academy that same month. In 1974, he graduated from the Air Force Academy with a Bachelor of Science degree in Astronautical Engineering and a commission in the United States Air Force. From 1974 to 1978, he served as satellite operations engineer for the Satellite Control Facility, SAMSO, at Sunnyvale Air Force Station, California. In 1978, Captain Harnly was assigned to the Air Force Institute of Technology to pursue a Master's Degree in Astronautical Engineering.

Permanent address: 4155 Hidden Circle
Colorado Springs, Colorado 80917

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GA/EE/79-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN ADAPTIVE DISTRIBUTED-MEASUREMENT EXTENDED KALMAN FILTER FOR A SHORT RANGE TRACKER		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert L. Jensen, Captain, USAF Douglas A. Harnly, Captain, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December 1979
		13. NUMBER OF PAGES (Vol I) (Vol II)
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17 Joseph P. Hipps, Major, USAF Director of Public Affairs, AFIT		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Kalman filter Adaptive estimation Target tracking Atmospheric jitter		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) An adaptive Extended Kalman Filter algorithm is designed to track a distributed (elliptical) source target in a closed loop tracking problem, using outputs from a forward looking infrared (FLIR) sensor as measurements. The filter adaptively estimates image intensity, target size and shape,		

dynamic driving noise, and translational position changes due to two effects: actual target motion, and atmospheric jitter. Atmospheric backgrounds are studied for the effect of temporal and spatial correlations on filter performance. A Monte Carlo analysis is conducted to determine filter performance for two target scenarios: approximately straight approach and cross range constant velocity. Good performance is obtained for the first two trajectories. For the second trajectory, a one sigma tracking error of .2 pixel (4 ^{micro}rad) with a signal to noise ratio of 12.5. The filter adapts well to changes in image intensity, size, and shape.